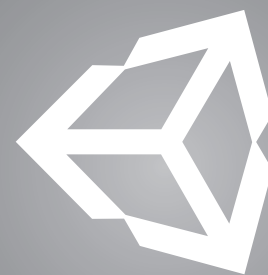


msdn

magazine



Build a Virtual Reality App with Unity.....10

Your Next Great Dashboard Starts Here

Create high impact and information rich decision support systems for desktops and the web with the DevExpress Universal Subscription.



Download Your Free 30-Day Trial
devexpress.com/dashboard



"A must have for any serious developer."

Ron Lindsey

**"DevExpress works as a major
workforce multiplier."**

Peter Van Zyl

**"One of the most powerful, feature rich
control suites on the market."**

Chris Todd

#UseTheBest

msdn magazine



Build a Virtual Reality
App with Unity.....10

Building Virtual Reality Applications Tim Kulp	10
The Source of Truth: The Role of Repositories in DevOps Kraig Brockschmidt	16
Feature Slices for ASP.NET Core MVC Steve Smith	26
Build Asynchronous AJAX-Enabled Web Pages with Reactive Extensions Peter Vogel	30
Cross-Platform Productivity with Xamarin Kevin Ashley	36
Unicode Encoding Conversions with STL Strings and Win32 APIs Giovanni Dicanio	44

COLUMNS

CUTTING EDGE

Message-Based Business Logic
in Practice
Dino Esposito, page 6

TEST RUN

The Secretary Problem
James McCaffrey, page 56

ESSENTIAL .NET

Command-Line Processing
with .NET Core 1.0
Mark Michaelis, page 62

MODERN APPS

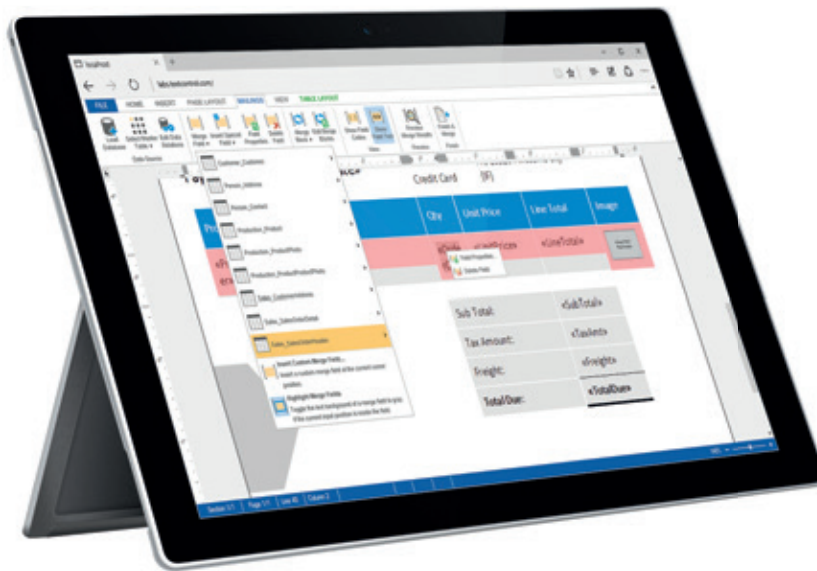
Build a City-Wide Wi-Fi Scanner
with UWP and Azure
Frank La Vigne, page 74

DON'T GET ME STARTED

Mary Jane Grows Up
David Platt, page 80

ASP.NET MVC REPORTING

The first cross-browser, true WYSIWYG HTML5-based document editor.
The Text Control Reporting Framework combines powerful reporting features with an easy-to-use, MS Word compatible word processor.

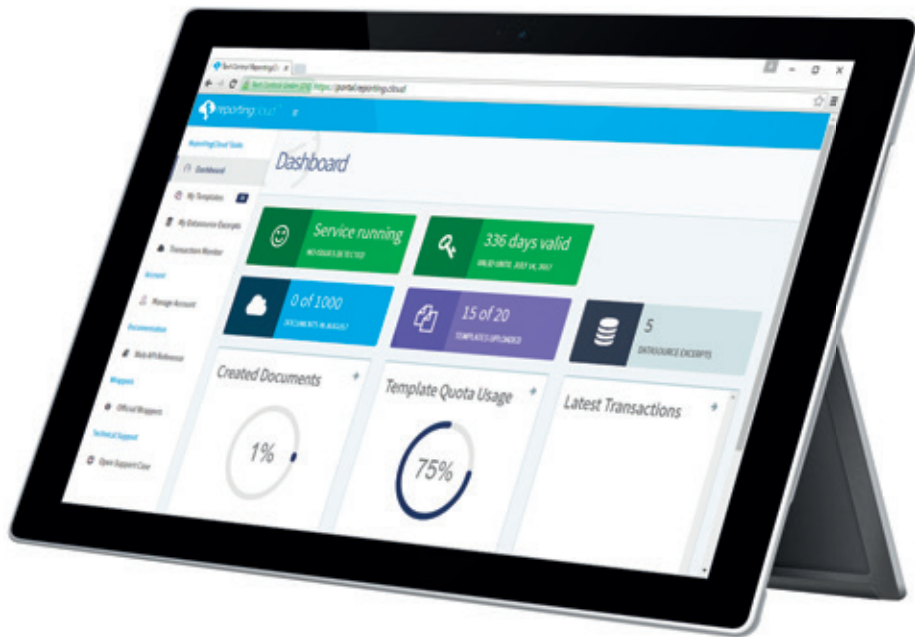


Give your users an MS Word compatible editor to create powerful documents and reporting templates anywhere - in any browser. Feature-complete including mail merge, sections, headers and footers, drawings and shapes, tables, barcodes and charts.

Live demo and 30-day trial version download at:
www.textcontrol.com/reporting

CREATE DOCUMENTS IN THE

Web API powered reporting platform to create
MS Word compatible reports in the cloud.



Use the Text Control ReportingCloud REST web service to create powerful,
MS Word compatible reports with JSON data from all clients such as
.NET, Java, JavaScript, Ruby, PHP, Python, Android, iOS and many more.
Templates can be created and edited directly in any browser.

Free trial account and wrapper documentation at:
www.reporting.cloud

General Manager Jeff Sandquist

Director Dan Fernandez

Editorial Director Mohammad Al-Sabt mmeditor@microsoft.com

Site Manager Kent Sharkey

Editorial Director, Enterprise Computing Group Scott Bekker

Editor in Chief Michael Desmond

Features Editor Sharon Terdeman

Features Editor Ed Zintel

Group Managing Editor Wendy Hernandez

Senior Contributing Editor Dr. James McCaffrey

Contributing Editors Dino Esposito, Frank La Vigne, Julie Lerman, Mark Michaelis, Ted Neward, David S. Platt, Bruno Terkaly

Vice President, Art and Brand Design Scott Shultz

Art Director Joshua Gould



President
Henry Allain

Chief Revenue Officer
Dan LaBianca

Chief Marketing Officer
Carmel McDonagh

ART STAFF

Creative Director Jeffrey Langkau

Associate Creative Director Scott Rovin

Senior Art Director Deirdre Hoffman

Art Director Michele Singh

Assistant Art Director Dragutin Cvijanovic

Graphic Designer Erin Horlacher

Senior Graphic Designer Alan Tao

Senior Web Designer Martin Peace

PRODUCTION STAFF

Print Production Coordinator Lee Alexander

ADVERTISING AND SALES

Chief Revenue Officer Dan LaBianca

Regional Sales Manager Christopher Kourtoglou

Account Executive Caroline Stover

Advertising Sales Associate Tanya Egenolf

ONLINE/DIGITAL MEDIA

Vice President, Digital Strategy Becky Nagel

Senior Site Producer, News Kurt Mackie

Senior Site Producer Gladys Rama

Site Producer Chris Paoli

Site Producer, News David Ramel

Director, Site Administration Shane Lee

Site Administrator Biswarup Bhattacharjee

Front-End Developer Anya Smolinski

Junior Front-End Developer Casey Rysavy

Executive Producer, New Media Michael Domingo

Office Manager & Site Assoc. James Bowling

LEAD SERVICES

Vice President, Lead Services Michele Imgrund

Senior Director, Audience Development & Data Procurement Annette Levee

Director, Audience Development & Lead Generation Marketing Irene Fincher

Director, Client Services & Webinar Production Tracy Cook

Director, Lead Generation Marketing Eric Yoshizuru

Director, Custom Assets & Client Services Mallory Bundy

Senior Program Manager, Client Services & Webinar Production Chris Flack

Project Manager, Lead Generation Marketing Mahal Ramos

Coordinator, Lead Generation Marketing Obum Ukabam

MARKETING

Chief Marketing Officer Carmel McDonagh

Vice President, Marketing Emily Jacobs

Senior Manager, Marketing Christopher Morales

Marketing Coordinator Alicia Chew

Marketing & Editorial Assistant Dana Friedman

ENTERPRISE COMPUTING GROUP EVENTS

Vice President, Events Brent Sutton

Senior Director, Operations Sara Ross

Senior Director, Event Marketing Merikay Marzoni

Events Sponsorship Sales Danna Vedder

Senior Manager, Events Danielle Potts

Coordinator, Event Marketing Michelle Cheng

Coordinator, Event Marketing Chantelle Wallace



Chief Executive Officer
Rajeev Kapur

Chief Operating Officer
Henry Allain

Vice President & Chief Financial Officer
Michael Rafter

Chief Technology Officer
Erik A. Lindgren

Executive Vice President
Michael J. Valenti

Chairman of the Board
Jeffrey S. Klein

ID STATEMENT MSDN Magazine (ISSN 1528-4859) is published monthly by 1105 Media, Inc., 9201 Oakdale Avenue, Ste. 101, Chatsworth, CA 91311. Periodicals postage paid at Chatsworth, CA 91311-9998, and at additional mailing offices. Annual subscription rates payable in US funds are: U.S. \$35.00, International \$60.00. Annual digital subscription rates payable in U.S. funds are: U.S. \$25.00, International \$25.00. Single copies/back issues: U.S. \$10, all others \$12. Send orders with payment to: MSDN Magazine, P.O. Box 3167, Carol Stream, IL 60132, email MSDNmag@1105service.com or call (847) 763-9560. POSTMASTER: Send address changes to MSDN Magazine, P.O. Box 2166, Skokie, IL 60076. Canada Publications Mail Agreement No: 40612608. Return Undeliverable Canadian Addresses to Circulation Dept. or XPO Returns: P.O. Box 201, Richmond Hill, ON L4B 4R5, Canada.

Printed in the U.S.A. Reproductions in whole or part prohibited except by written permission. Mail requests to "Permissions Editor," c/o MSDN Magazine, 4 Venture, Suite 150, Irvine, CA 92618.

LEGAL DISCLAIMER The information in this magazine has not undergone any formal testing by 1105 Media, Inc. and is distributed without any warranty expressed or implied. Implementation or use of any information contained herein is the reader's sole responsibility. While the information has been reviewed for accuracy, there is no guarantee that the same or similar results may be achieved in all environments. Technical inaccuracies may result from printing errors and/or new developments in the industry.

CORPORATE ADDRESS 1105 Media, 9201 Oakdale Ave. Ste 101, Chatsworth, CA 91311 www.1105media.com

MEDIA KITS Direct your Media Kit requests to Chief Revenue Officer Dan LaBianca, 972-687-6702 (phone), 972-687-6799 (fax), dlabianca@1105media.com

REPRINTS For single article reprints (in minimum quantities of 250-500), e-prints, plaques and posters contact: PARS International Phone: 212-221-9595. E-mail: 1105reprints@parsintl.com. www.magreprints.com/QuickQuote.asp

LIST RENTAL This publication's subscriber list, as well as other lists from 1105 Media, Inc., is available for rental. For more information, please contact our list manager, Jane Long, Merit Direct. Phone: 913-685-1301; E-mail: jl@meritdirect.com; Web: www.meritdirect.com/1105

Reaching the Staff

Staff may be reached via e-mail, telephone, fax, or mail. A list of editors and contact information is also available online at Redmondmag.com.

E-mail: To e-mail any member of the staff, please use the following form: FirstInitial.LastName@1105media.com
Irvine Office (weekdays, 9:00 a.m. – 5:00 p.m. PT)
Telephone 949-265-1520; Fax 949-265-1528
4 Venture, Suite 150, Irvine, CA 92618

Corporate Office (weekdays, 8:30 a.m. – 5:30 p.m. PT)
Telephone 818-814-5200; Fax 818-734-1522
9201 Oakdale Avenue, Suite 101, Chatsworth, CA 91311

The opinions expressed within the articles and other contents herein do not necessarily express those of the publisher.



THE PREMIER **BARCODE** SDK

LEADTOOLS Barcode SDK is the world's premier toolkit for developing applications that require reading and writing **1D** and **2D barcodes**. With comprehensive support of over 100 different barcode types and sub-types such as UPC, EAN, Code 128, QR Code, Data Matrix, PDF417, Aztec, Maxicode, and Micro QR, LEADTOOLS is the right choice.

.NET Windows API Java WinRT Linux iOS OS X Android JavaScript





Our Virtual Future

Pokemon Go is destroying our civilization. People are falling off cliffs (lat.ms/2acQAYm), ramming police cars (usat.ly/2a2CHDd), driving into trees (usat.ly/29QjW2x) and even getting *stuck* in trees (ti.me/29Wzym4)—all in pursuit of little virtual creatures that can only be found within the confines of a smartphone display.

You might call this kind of behavior madness. I call it the future.

As ever, past is prologue. From the moment the first Sony Walkman user clapped a pair of foam earphones on his head only to look up a minute later to find an angry parent trying to get his attention, this was inevitable. We have over the decades retreated from our environments. Portable cassette and, later, MP3 players isolated us from our surroundings, before smartphones gave us the ability to interact beyond them. As *MSDN Magazine* columnist David Platt observed in his February 2012 Don't Get Me Started column, "as technology closes the distance between people around the world, it simultaneously creates distance between people in close proximity" (msdn.com/magazine/hh781031).

In a sense, Pokemon Go, and augmented reality in general, is closing the loop on this trend, returning users to manipulated versions of their local environments. Pokemon Go players are blundering into physical objects because they are chasing things that *are not there*. They are at once present and removed, assuming a quantum state that is at odds with our Newtonian understanding of the social world.

We'll learn to manage the dichotomy, eventually. Tim Kulp is the author of this month's feature article, "Building Virtual Reality Applications." He says virtual, augmented and mixed reality (VAMR) applications will require the same process of societal integration that occurred with Walkmans and mobile phones, as it did with televisions, radios and cars before that.

"Is it polite to be on the phone while checking out at the grocery store? These are things we didn't just know. We learned over time,"

Kulp says. "VAMR is the same. Is it OK to walk into a board room meeting wearing a HoloLens? Social norms need to be figured out, and to do that we first need to challenge those norms."

There's work to be done yet. Kulp says the tools and platforms to enable compelling VAMR applications must mature and proliferate. Most important, the sector needs apps that matter.

From the moment the first Sony Walkman user clapped a pair of foam earphones on his head only to look up a minute later to find an angry parent trying to get his attention, this was inevitable.

"I think one of the challenges that technologists often face is the belief that technology is about building cool things that make life better," Kulp says. "In reality, technology is about society, change and giving new tools to people to do things. When we forget about the social element of technology, we don't build anything useful. One challenge to VAMR right now is the 'figuring out' of its place in society."

Pokemon Go is an early, clumsy expression of our coming augmented, social future. I look forward to seeing where these first, halting steps take us.

Visit us at msdn.microsoft.com/magazine. Questions, comments or suggestions for *MSDN Magazine*? Send them to the editor: mmeditor@microsoft.com.

© 2016 Microsoft Corporation. All rights reserved.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, you are not permitted to reproduce, store, or introduce into a retrieval system *MSDN Magazine* or any part of *MSDN Magazine*. If you have purchased or have otherwise properly acquired a copy of *MSDN Magazine* in paper format, you are permitted to physically transfer this paper copy in unmodified form. Otherwise, you are not permitted to transmit copies of *MSDN Magazine* (or any part of *MSDN Magazine*) in any form or by any means without the express written permission of Microsoft Corporation.

A listing of Microsoft Corporation trademarks can be found at microsoft.com/library/toolbar/3.0/trademarks/en-us.mspx. Other trademarks or trade names mentioned herein are the property of their respective owners.

MSDN Magazine is published by 1105 Media, Inc. 1105 Media, Inc. is an independent company not affiliated with Microsoft Corporation. Microsoft Corporation is solely responsible for the editorial contents of this magazine. The recommendations and technical guidelines in *MSDN Magazine* are based on specific environments and configurations. These recommendations or guidelines may not apply to dissimilar configurations. Microsoft Corporation does not make any representation or warranty, express or implied, with respect to any code or other information herein and disclaims any liability whatsoever for any use of such code or other information. *MSDN Magazine*, MSDN and Microsoft logos are used by 1105 Media, Inc. under license from owner.



Create and Edit PDFs in .NET, COM/ActiveX, WinRT & UWP

NEW
v5.5

- Edit, process and print PDF 1.7 documents
- Create, fill-out and annotate PDF forms
- Fast and lightweight 32- and 64-bit components for .NET and ActiveX/COM
- New Universal Apps DLLs enable publishing C#, C++, CX or Javascript apps to windows Store
- Updated Postscript/EPS to PDF conversion module



Complete Suite of Accurate PDF Components

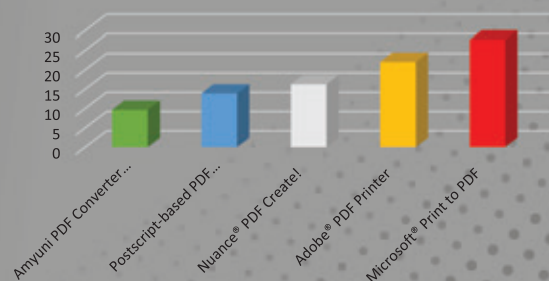
- All your PDF processing, conversion and editing in a single package
- Combines Amyuni PDF Converter and PDF Creator for easy licensing, integration and deployment.
- Includes our Microsoft WHQL certified PDF Converter printer driver
- Export PDF documents into other formats such as Jpeg, PNG, XAML or HTML5
- Import and Export XPS files using any programming environment



High Performance PDF Printer for Desktops and Servers

- Print to PDF in a fraction of the time needed with other tools. WHQL tested for all Windows platforms. Version 5.5 updated for Windows 10 support

Benchmark Testing - Amyuni vs Others
Seconds required to convert a document to PDF



Other Developer Components from Amyuni®

- WebkitPDF: Direct conversion of HTML files into PDF and XAML without the use of a web browser or a printer driver
- PDF2HTML5: Conversion of PDF to HTML5 including dynamic forms
- Postscript to PDF Library: For document workflow applications that require processing of Postscript documents
- OCR Module: Free add-on to PDF Creator uses the Tesseract engine for character recognition
- Javascript engine: Integrate a full Javascript interpreter into your applications to process PDF files or for any other need



AMYUNI 
Technologies

USA and Canada

Toll Free: 1866 926 9864

Support: 514 868 9227

sales@amyuni.com

Europe

UK: 0800-015-4682

Germany: 0800-183-0923

France: 0800-911-248

All trademarks are property of their respective owners. © Amyuni Technologies Inc. All rights reserved.

All development tools available at

www.amyuni.com



Message-Based Business Logic in Practice

It's common knowledge that to create efficient business software, you need to carefully model it around business events and faithfully mirror business processes.

Some 20 years ago, the flowchart used to be the ideal tool to represent large chunks of an application. Over the years, the size of flowcharts grew so big to become impractical to handle and engineers started looking elsewhere to find an appropriate model to outline the complexity of the business. Comprehensive, all-inclusive Unified Modeling Language (UML) diagrams appeared to be the way to go, starting from the class diagram, which is used to convey the domain model. That seemed to be the essence of Domain-Driven Design (DDD) and it worked for many projects, but also failed for reasons including poor design skills, lack of communication, inadequate range of technologies and, above all, insufficient understanding of the mechanics of the business processes.

Put together, the strategies
“mirroring the business processes”
and “mirroring the users’
procedures” make up for quite a
different way of designing software.

An emerging and more modern approach to software engineering is to look at business events and use such events to gather requirements and build a deeper understanding of the mechanics of the system. In other words, instead of artificially creating a new model for the observed business, you aim at mirroring through code the same business processes that stakeholders and analysts describe and doing so using the procedures with which end users are familiar.

In this column, I'll provide a practical demonstration of this approach by expressing relevant business processes through software entities called “sagas.” In addition, I'll use ad hoc view models to populate each screen with which users are familiar.

Finally, I'll break up the steps of each application workflow into a combination of messages (commands and events) exchanged by involved parties. You'll be surprised how easier it becomes expressing even complex pieces of business logic and, more than anything else, how quick and reliable it can be changing the implementation to stay in sync with variable business rules.

I'll introduce a bunch of new software entities such as sagas, commands, events, buses and application services. To not reinvent the wheel for any new application—and to save a great deal of repetitive code—I'll also sketch out the characteristics of a framework that can help you implement this kind of event-based architecture.

Alternative Way of Achieving the Same

Put together, the strategies “mirroring the business processes” and “mirroring the users' procedures” make up for quite a different way of designing software. The overall methodology becomes top-down rather than bottom-up. A viable way could be to start designing the system from the presentation layer, which is primarily driven by the view models acting behind the user-facing screens and wizards. Altogether, users enter information in screens, which forms the view model of the screen and becomes input for any subsequent business actions to be taken against the system.

As users work with the UI, commands are pushed to the application layer—the topmost substrate of the business logic in charge of capturing any user input and returning any desired views. The application layer is seen as the orchestrator of the logic behind all the use cases of the application. Typically, the application layer is implemented as a set of services so a class library populated with methods that are one-to-one with user actions will definitely do the job. In the context of an ASP.NET MVC application, an

application layer class can be one-to-one with controller classes and each controller method that triggers a business operation ends up calling into a particular application layer method in a one-to-one manner, as shown in **Figure 1**.

Next, I'll show you how to rewrite a typical business action such as registering an invoice using a message-based approach.

From Presentation and Back

As a reference, I'll use an excerpt of the sample application that Andrea Saltarello

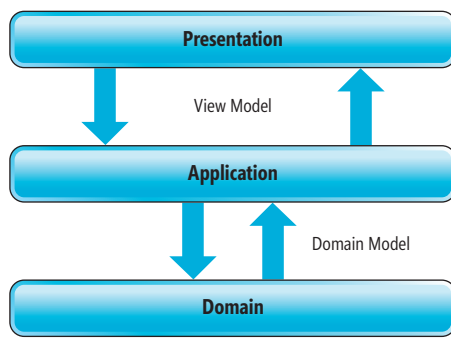
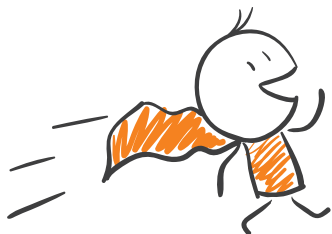
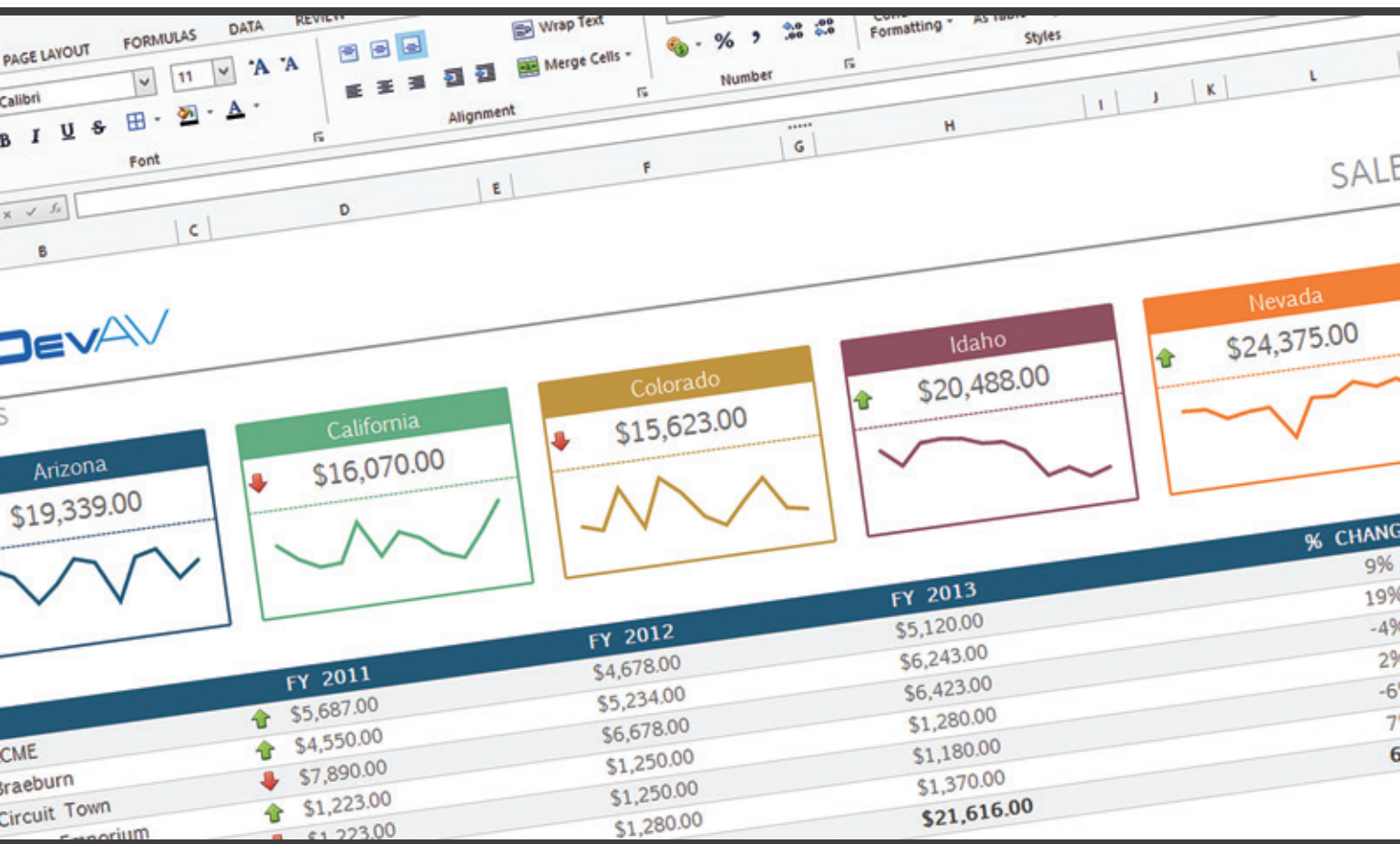


Figure 1 View Model and Application Layer

Office® Inspired Apps

Get started today and create high-performance, high-impact .NET solutions that fully replicate the look, feel and user-experience of **Microsoft Office.®**



Your Next Great Business App Starts Here

Explore our complete range of Office-inspired controls for all major .NET platforms.

devexpress.com/office



and I originally wrote as companion code for our Microsoft Press architecture book, “Microsoft .NET—Architecting Applications for the Enterprise, 2nd Edition” (2014). In the past year, the code evolved significantly and you can find it at bit.ly/29Nf2aX under the MERP folder.

MERP is an ASP.NET MVC application articulated in two areas. Each area is close to being a bounded context in the jargon of DDD and each can be considered, more generically, a microservice. In ASP.NET MVC, an area is a distinct collection of controllers, models and Razor views. Say, then, you have an InvoiceController class with an Issue method. Realistically, the Issue action is implemented through distinct GET and POST methods, as shown in **Figure 2**.

In a classic implementation scenario, the worker service method will collect input data and trigger a sequential hardcoded workflow.

The member WorkerServices is the entry point in the application layer and is a reference injected in the controller class through dependency injection:

```
public InvoiceController(InvoiceControllerWorkerServices workerServices)
{
    if(workerServices == null)
        throw new ArgumentNullException(nameof(workerServices));

    WorkerServices = workerServices;
}
```

The structure of the GET method is fairly simple: The application layer retrieves the view model that contains all the data necessary to present the user screen to trigger the “Issue” action. The data the worker service returns is passed as is to the Razor view.

The POST method follows a CQRS logic and executes the command that will issue the invoice, thus altering the state of the system. The command doesn't necessarily produce any direct feedback for the user, but a redirect occurs to physically separate the command from any successive queries that will actually update the UI.

Breaking up the Necessary Logic

In a classic implementation scenario, the worker service method will collect input data and trigger a sequential hardcoded workflow. At the end of the workflow, the service method will pick up results and package them into a data structure to be returned to upper layers of code.

The workflow is typically a monolithic procedure made of conditional branches, loops and whatever serves to express the required logic. The workflow is likely inspired by a flowchart diagram drawn by domain experts, but at the end of the day it turns out to be a software artifact in which the flow of the work is flattened through the constructs and intricacies of the programming language or ad hoc workflow frameworks. Making even a small change might

have significant regression effects. While unit tests exist to control regression, they're still often perceived as an extra burden, requiring extra costs and effort.

Let's see what happens, instead, if you orchestrate the business processes using commands.

Pushing Commands to the Bus

In message-based software terms, a command is a packet of data, namely a POJO class with only properties and no methods. In abstract terms, instead, a command can be described as an imperative message delivered to the system to have some tasks performed. Here's how you can trigger a task by pushing a command to the system:

```
public void Issue(IssueViewModel model)
{
    var command = new IssueInvoiceCommand(
        model.Date,
        model.Amount,
        ...
    );
    Bus.Send(command);
}
```

The view model contains all the input data for the system to process. The model binding layer of ASP.NET MVC already does good mapping work between posted data and command data. It would come as no surprise if you were thinking of using the command class as the target of ASP.NET MVC model binding. In practice, though, model binding takes place in the controller context which is part of the presentation layer, whereas a command is a message that follows any decision made at the application layer level or even later. Admittedly, in the previous code snippet, the command class is close to being a copy of the view model class, but that's mostly due to the extreme simplicity of the example.

Once you have a command instance ready, the next step is delivering it to the system. In a classic software design, a command has an executor that takes care of it from start to finish. The difference here is that any command can be rejected by the system and, more important, the effect of the command, as well as the list of actual handlers, might be different depending on the state of the system.

A bus is a publish/subscribe mechanism that's primarily responsible for finding a handler for the command. The effect of the handler can be a single atomic action or, more likely, the command can trigger a saga. A saga is an instance of a known business process that typically comprises multiple commands and notifications of events for other sagas and handlers to which to react.

Figure 2 GET and POST Methods for a Controller Action

```
[HttpGet]
public ActionResult Issue()
{
    var model = WorkerServices.GetIssueViewModel();
    return View(model);
}

[HttpPost]
public ActionResult Issue(IssueViewModel model)
{
    if (!this.ModelState.IsValid)
    {
        return View(model);
    }
    WorkerServices.Issue(model);
    return Redirect("/Accountancy/");
}
```


The bus is a central element in a message-based architecture. At the very minimum, the bus interface is likely to include the following methods:

```
public interface IBus
{
    void Send<T>(T command) where T : Command;

    void RegisterSaga<T>() where T : Saga;

    void RegisterHandler<T>();
}
```

In addition to the Send method used to publish commands, the bus typically offers methods to register sagas and handlers. Sagas and handlers are similar software components in the sense that both handle commands. A handler, however, starts and finishes in a single run without ever getting back to the bus. A common type of handler is the denormalizer, namely a component that saves a read-only projection of the current state of an aggregate in a full CQRS architecture. A saga can be a multi-step process made of multiple commands and event notifications pushed back to the bus for other handlers and sagas to react.

Configuring the Bus

Configuration of the bus takes place during the application's startup. At this time, business processes are split in sagas and optionally handlers. Each saga is fully identified by a starter message (command or event) and a list of messages it will handle. **Figure 3** is an example.

A saga class typically inherits from a base class that conveniently packages references to assets you're likely going to use such as the event store, the bus and the aggregate repository. As mentioned, a saga is also characterized by the starter message summarized by the `IAmStartedBy<T>` interface where T is either an event or a command:

```
public interface IAmStartedBy<T>
{
    void Handle(T message);
}
```

As you can see, the `IAmStartedBy<T>` interface counts a single method—Handle—which gets an instance of the type T. The body of any Handle method contains the actual code to register the invoice and, more in general, to perform the business task. At the end of the task, there might be the need of notifying other sagas or handlers of what happened. In particular, you might want to raise an event to let others know that the invoice was registered successfully or whether it failed and why. Handlers of a “failed” notification will

Figure 3 Configuring the Bus

```
// Sagas
var bus = new SomeBus();
...
bus.RegisterSaga<IncomingInvoiceSaga>();
bus.RegisterSaga<OutgoingInvoiceSaga>();

// Handlers (denormalizers)
bus.RegisterHandler<IncomingInvoiceDenormalizer>();
bus.RegisterHandler<InvoiceDenormalizer>();
bus.RegisterHandler<OutgoingInvoiceDenormalizer>();
// Here's a skeleton for the saga components above.
public class IncomingInvoiceSaga : Saga,
    IAmStartedBy<RegisterIncomingInvoiceCommand>
{
    ...
}
public class OutgoingInvoiceSaga : Saga,
    IAmStartedBy<IssueInvoiceCommand>
{
    ...
}
```

then be responsible for any compensation or rollback procedures and for generating any sort of feedback to the user.

From Flowcharts to Sagas

There's an inherent similarity between flowcharts that domain experts may be able to use to outline a business process and how you define sagas in your code. In a way, the saga is the implementation of a flowchart where each block may be rendered through a received event or command to handle. A saga can be a long-running process and can even be suspended and resumed. Think, for example, of a business process that comprises an approval step (such as after doing offline research on a customer). The saga must be started and when the stage of approval is reached it should receive an event and get serialized by the handler. Next, the saga will be resumed when some code sends the command that witnesses the approval. As you can see, having the business logic split in micro steps also makes it easier to extend and alter the logic to stay in sync with the evolving business.

In addition to the Send method used to publish commands, the bus typically offers methods to register sagas and handlers.

From Theory to Frameworks

So far I assumed the existence of components such as sagas and buses. Who writes the bus and deals with persistence of sagas? And what about the definitions of interfaces and base classes that I mentioned as part of a framework?

If you look at the source code of project MERP from bit.ly/29Nf2aX, you'll see that it uses classes like Bus, Saga, Command and much more. In the latest version of the code, though, these classes come from a few new NuGet packages, collectively known as the MementoFX. A typical application based on the Memento framework requires the core MementoFX package, which defines base classes and a couple of helper packages for the bus (Memento.Messaging.Postie or Memento.Messaging.Rebus presently) and event persistence. At the moment, the MERP source code uses the Embedded RavenDB engine for persistence and wraps it up in a Memento.Persistence.EmbeddedRavenDB package. This way, the event store will be started as soon as the ASP.NET process is up.

By using the MementoFX packages, you can dramatically reduce the effort to build message-based business logic. Let me hear your thoughts! ■

DINO ESPOSITO is the author of “Microsoft .NET: Architecting Applications for the Enterprise” (Microsoft Press, 2014) and “Modern Web Applications with ASP.NET” (Microsoft Press, 2016). A technical evangelist for the .NET and Android platforms at JetBrains, and frequent speaker at industry events worldwide, Esposito shares his vision of software at software2cents@wordpress.com and on Twitter: @despos.

THANKS to the following Microsoft technical expert for reviewing this article: Andrea Saltarello

Building Virtual Reality Applications

Tim Kulp

Virtual reality (VR) provides a unique opportunity and challenge for application developers. On one hand, there's opportunity to build immersive experiences unbound by the existing 2D UX paradigm. On the other hand, there's the challenge to build a great VR experience that's more than a 3D view of a 2D screen. Fortunately, some fundamentals of app development and UI design are similar, such as creating a master/details view, getting input from the user, and communicating across applications via services. In this article, I'll explore these topics from the context of building a business application in VR so that you can see how to start your own VR apps.

Why VR?

It's a good time to enter VR app development with the availability of great tools such as Unity and the relatively small VR marketplace.

This article discusses:

- Building a basic UI in Virtual Reality
- Connecting to external data in a Virtual Reality application
- Data binding game objects to data in Unity

Technologies discussed:

Unity, C#, Virtual Reality

Code download available at:

bit.ly/2aRTlo4

Think about the current mobile app market space. There are millions of apps and new apps have a lot of competition. The VR market space is still small, however, making it easier to stand out and get noticed. On top of the market size, there's a great hunger currently for VR content with early adopters. How to build apps in this space is still a new challenge, with best practices evolving. Early adopters in VR development will help shape the future of this platform. It's an exciting time to be in VR.

Now's a great time to get into VR app development because of the availability of great tools such as Unity and the relatively small marketplace.

Building apps for VR devices such as Samsung Gear VR and Google Cardboard is a great entry point for developers to start building their content for low-cost distribution channels. If built with a mind toward Mixed Reality, VR apps can be ported to the HoloLens because the tools are the same and the content can be the same. The only difference is the code to perform tasks (such as

Gaze vs. Raycast, which I will explore later in this article), and utilizing some of the HoloLens unique features such as speech, spatial mapping and gesture. With a little research and experimentation, you can convert your VR app to a HoloLens app.

I won't go into exhaustive detail on setting up a VR project. Instead, I recommend exploring the Unity VR Samples project in the Unity Asset Store (bit.ly/1qYLBX9). The project built in this article will start from scratch, but will use scripts from the Unity VR Samples project to leverage existing code for standard tasks (such as making an object interactive). I recommend creating a blank Unity project and importing the Unity VR Samples for reference outside of the project that'll be built with this article. Unity VR Samples has a lot of great content but is built to be a standalone app and requires some extra configuration for running a scene outside the included scenes. Keep Unity VR Samples open and available because it has a few scripts that will be used in this project. For this article, I'll use Unity 5.3 with the Android tools enabled so that I can build the app for the Samsung Gear VR device.

Contoso Travel

Contoso Travel is a corporate travel system that lets a travel administrator see the bookings for their team members. Travel administrators can view travelers by airport, search for a specific city, and view a traveler's trip details. In the app, I want to create a rich experience that shows the spatial relationship of travelers for Contoso. Normally this would be a line-of-business app, mobile or Web, delivered to the users through a browser. For this article, I'll build this app in VR to give users a deep experience, use avatars to represent the travelers and show where people are in relation to each other within 3D space.

VR opens many possibilities for an app such as this. From the travel administrator's perspective, VR can be a more personal experience, engaging the traveler's avatars instead of just displaying dots on a map or a name on a list. For travelers, VR can give travelers a preview of the office they're visiting or sites to see while in a remote location. The idea here is to create an immersive experience that users can enjoy.

Starting Out

You'll want to prepare this project as a mobile VR experience. To start, create a new 3D Unity Project called ContosoTravelVR. First, change the timing of the world to a slower framerate. In the Edit | Project Settings | Time inspector, set Fixed Timestep = 0.01666666, set Maximum Allowed Timestep = 0.1666666 and Timescale = 1. Maximum Allowed Timestep slows down Unity's physics time so that physics calculations do not impact framerate. While physics time and framerate are not directly related, the CPU power required for physics calculation can create issues with processing frames. Setting Maximum Allowed Timestep puts a maximum limit on how much time to spend on physics updates. Remember that in a mobile VR world, the device is drawing two images every frame: one for the left eye and one for the right eye. Complex computations processing at higher frame rates can cause a device, such as a phone, to get frames out of sync, causing each eye to see something different. That causes disorientation and nausea in users.

Obviously, this is critical to avoid for a great user experience. Avoid anything that could have a performance impact, such as high polygon counts for objects, long-running scripts or locking the thread.

Next, update the Player Settings for mobile VR. To do this, go to Edit | Project Settings | Player and select the Other Settings section. Within this section there's a setting called Virtual Reality Supported. Check the box to ensure that the world is recognized as a VR app. This setting will automatically convert the Main Camera object that already exists in Unity to a VR camera.

For travelers, VR can give
travelers a preview of the office
they're visiting or sites to see
while in a remote location.

Now that the app is set, set up the scene. In the Hierarchy window, add a cube with the scale of x=10, y=0.1 and z=10. Reset the position of the cube to be x=0, y=0 and z=0 to center the flattened cube in the world. This will provide a floor for your world, which gives users a sense of space and grounding, as well as a place for the entities in the world on which to stand. Just like in 2D apps, you need to give the user a sense of where the content and interactions of the app will occur. This helps users orient themselves in the app. For ContosoTravel, all actions will occur within the bounds of this floor object.

In this world, you need an object that will marshal data throughout the app. Create an Empty Game Object in the Hierarchy window and rename the object to TravelerManager. This object will be the source of all the travel data within the app. While selecting the TravelerManager game object, click Add Component | New Script. Name the script TravelerManager and click Create and Add. This will create a new script to handle the logic of the TravelerManager.

As a travel app, you need to see travelers in the world. For this, add a new Game Object to the Hierarchy and call it TravelerTemplate. This will be the object that's shown for each traveler represented in the system. If you're familiar with prefabs, use a character you've designed or something from the Asset Store. For my app, I'll use some low-poly

Figure 1 TravelerManager Script Component

```
public GameObject TravelerTemplate;
public List<Traveler> TravelerList;

void Awake () {

    TravelerList = new List<Traveler>();
}
public void LoadTravelers() {
    try
    {
        // Add GetTravelers here
    }
    catch(Exception ex)
    {
        Debug.Log(ex.ToString());
    }
}
```

characters I found in the Asset Store that'll give the world a fun, blocky look. Any game object will work for the `TravelerTemplate`—a prefab, a cube, a sphere and so on. Once the `TravelerTemplate` is added, disable the game object by clicking on the checkbox next to the game object's name in the Inspector window. This will make the `TravelerTemplate` disappear from the screen.

Finally, add the following scripts from the Unity VR Samples project to your project: `VREyeRaycast`, `VRInput`, `VRInteractiveItem` and `Reticle`. I'll discuss the exact purpose of these scripts as I use them throughout the article, but they'll save me a lot of coding later in the app.

Setting Up the TravelerManager Class

In Unity, double-click the `TravelerManager` script to open it in Mono-Develop or Visual Studio. Add the code in **Figure 1** to the component.

The `TravelerTemplate` will hold the Prefab I'm using to create the travelers on the screen. Remember, `TravelerTemplate` can be any game object: a cube, a sphere, a prefab character and so on. Also, by marking this variable public, the value can be set in the Unity design tools, which let the design team control the traveler template without needing to re-code the `TravelerManager`. You also create two lists, one for storing all the travelers created as `GameObjects`, and the other to maintain the data for each traveler. Next, you'll add the input mechanisms to load the travelers and give users something to do in the application.

Hooking Up a Button

With the basic world set up, you need to collect input from the user to get the app doing stuff. To start, add a button that will trigger the `LoadTravelers` method of the `TravelerManager` script component. Why not just load the travelers by default? In this case, you want to provide a point of orientation for the user to understand what's happening in the app. VR is too new for users to have pre-defined expectations of what's going to happen as soon as they load a world. UI instructions are a good tool to help the user orient and acclimate to the new world.

Getting data from a URL is very simple in Unity, allowing apps to build out data-rich experiences.

Create a new Empty Game Object on the root of the application and call it GUI. This game object will hold the UI components for the app. With the GUI game object selected, add a new Canvas object. The Canvas object holds all UI components on the screen. Once the Canvas object is created, go to the Inspector and change the Render Mode to World Space. This will detach the canvas from the camera and let content be placed in the world, as opposed to stuck to the camera. Now add a new Button object to the Canvas. Set the button's text to "Load Travelers." Change the button's display transform to $x=1.87$, $y=.05$, $z=0$ and width = 9; height = 3 with

a scale of $x=0.25$, $y=0.25$, which makes the button smaller and in the view of the user.

Select the button object in the Hierarchy window and then select the plus sign on the On Click list (in the Inspector window) to add a new click event. Buttons can have as many click events as needed, which lets you use a single button to trigger many different events. This is good for triggering an action and then disabling or hiding the button. Each click event requires a game object and a function available to that game object to be defined. After adding a click event, drag the `TravelerManager` game object to the Object field for the click event. Once the `TravelerManager` is set, the Function list will show the `LoadTravelers` method. If the function list doesn't show `LoadTravelers`, you need to build your code project in Visual Studio and try again to select `LoadTravelers`. Select the plus sign on the click event list again to add another event. Drag the button object from the Hierarchy to the Object field for the new click event, select `GameObject.isActive`, and leave the checkbox empty. This means that when the button is clicked, Unity will first execute `LoadTravelers` and then set the `isActive` value of the button to false to hide the button. From a user perspective, this button provides initial instructions. Once those instructions are complete, the instructions are no longer needed so you remove them.

Getting Data from a Service

Modern apps get their data from many different places. In this app, traveler data is provided by the `travel.contoso.com` API. Getting data from a URL is very simple in Unity, letting apps build out data-rich experiences. At the bottom of the `TravelerManager` script, add a new class called `Traveler`. This will be your data object from the traveler service:

```
public class Traveler{
    public int Id { get; set; }
    public string Name { get; set; }
    public string Title { get; set; }
    public string DepartureCity { get; set; }
    public string ArrivalCity { get; set; }
}
```

Now, you need to get data to fill this class. Create a new method under `LoadTravelers` called `GetTravelers`. This method will connect to the API endpoint, download the JSON object provided by the API and then send the result to another method to show the data request results on screen:

```
private IEnumerator GetTravelers(){
    var www = new WWW("https://travel.contoso.com/api/Traveler");
    yield return www;
    if (www.isDone)
        DisplayTravelers(www.text);
}
```

Now replace the "add `GetTravelers` here" comment in `LoadTravelers` with the following line of code:

```
StartCoroutine(GetTravelers());
```

`LoadTravelers` will call `GetTravelers`, which pulls the data from the API. The data from the API is sent to the `DisplayTravelers` method for further processing. To start this, use `StartCoroutine`, which executes `GetTravelers` without locking the thread and essentially freezing the app until execution is complete. Creating this smooth flow of code is one of the steps you take to ensure that the app doesn't drop frames. Coroutines start and pause a method with each frame, letting a method span multiple frames without locking the app. `StartCoroutine` is very

useful to do methods that could be long-running, such as waiting for data to return from a Web service. Coroutines can also be used for methods on large numbers of objects. For example, if the traveler service returned hundreds of people traveling—coroutines could be useful for calling actions across all the people, perhaps to emulate the appearance of wandering for the travelers. For more information on coroutines, see the Unity documentation at bit.ly/2ahxSBu.

In `GetTravelers` the `WWW` object from Unity is used to connect to the provided URL. Using the `yield` keyword, the `WWW` object is returned until it's complete by checking the `isDone` property. Once the download from the URL is complete, `isDone` will equal `true`, which will then pass the text returned by the API call to the `DisplayTravelers` method. The `WWW` object can be used to post data, as well as get data, so that the app could have functionality to add travelers in the future. Full documentation for the `WWW` object on the Unity site is at bit.ly/2aIf0ML.

Then you take the result from the API and convert it to something the user will be able to see in the app. Before you can show travelers in different locations on the map, you need to create those locations in the world. Go back to the Unity designer and create an empty Game Object on the root of the Hierarchy tree. Call this empty game object Location Container. Set Location Container to be at position `X=0, Y=0, Z=0` (center of the world). With the Location Container selected, create a few empty Game Object children scattered around the floor cube created when the article started. These empty Game Objects represent airports; the floor is like a map. In the `DisplayTravelers` method, you look at the traveler's destination city and match it to one of these empty Game Objects. Rename the empty Game Objects to some airport names. I used `BWI`, `MSP` and `SFO`. To complete the setup of airports, you create a tag for the Game Objects called `Airport` so that you can find all the airport items by searching for a single tag value. Creating a tag for a Game Object is done by going to the top left of the Inspector for the object, clicking on the tag dropdown list and selecting `New Tag`. You can create the new tag while all empty Game Objects are selected, which lets you set the tag for all the objects at once.

Add the code shown in **Figure 2** after the `getTravelers` method.

Figure 2 Convert JSON Traveler Objects to Unity Game Objects

```
private void DisplayTravelers(string json){
    var jsonTravelers = new JSONObject(json);
    foreach (var jsonTraveler in jsonTravelers.list){
        // Convert jsonobject to traveler object
        Traveler traveler = ConvertJSON(jsonTraveler);
        GameObject travelerRep =
            Instantiate(TravelerTemplate,
                new Vector3(0f, 0.05f, 0f),
                TravelerTemplate.transform.rotation) as GameObject;
        travelerRep.name = traveler.Name;

        GameObject[] airports = GameObject.FindGameObjectsWithTag("Airport");
        foreach (var airport in airports) {
            if (airport.name == traveler.ArrivalCity) {
                travelerRep.transform.position =
                    new Vector3(airport.transform.position.x, 0.05f,
                        airport.transform.position.z);
                break;
            }
        }
        TravelerList.Add(traveler);
    }
}
```



dtSearch®

Instantly Search Terabytes of Text

dtSearch's document filters support popular file types, emails with multilevel attachments, databases, web data

Highlights hits in all data types; 25+ search options

With APIs for .NET, Java and C++. SDKs for multiple platforms. (See site for articles on faceted search, SQL, MS Azure, etc.)

Visit dtSearch.com for

- hundreds of reviews and case studies
- fully-functional enterprise and developer evaluations

The Smart Choice for Text Retrieval® since 1991

dtSearch.com 1-800-IT-FINDS

Using the free JSONObject script from the Unity Asset store (bit.ly/2akN9p9), you're able to convert the JSON text to a Traveler C# code object. From there, using the Instantiate method, you create a copy of the TravelerTemplate, set it to the default coordinates, and name the Game Object the name of the traveler. This name is important for later in the app as it will be used to identify the traveler from the traveler list. Next, you find all the Game Objects tagged as Airports. This provides a collection of Airport objects that you can go through and find the matching object by name to place the traveler at the corresponding airport location. Using the tags you set up earlier lets you easily find the Airport Game Objects. At the end of the method, you add the traveler object to the TravelerList so that you can reference the traveler data later in the app. Now you have a master list of travelers displayed in your world. It's time to let users select a traveler so that you can show the detail information for that traveler.

Selecting a Traveler

Thinking about the world that's been created so far, a great way to do a details view is to have a speech bubble come from the avatar like the traveler is saying where they're traveling to. This would be a fun way to show the details while providing a UI in close proximity to the selected traveler.

In VR, selecting a traveler is done by looking at the object. To know if the user is looking at a traveler you need to use the Raycast object. Raycast is an invisible beam that emanates from the camera and lets the system know when an object with a collider component on it intersects with the beam. Using a Raycast lets your code detect when the user is looking at an object that should provide an action. Earlier you imported the VREyeRaycast script, which implements all the logic necessary to create a Raycast on a camera. Add the VREyeRaycast script to the Main Camera. You'll also need to add the Reticle script to the Main Camera to provide a UI component to show where the user is looking.

Using a Raycast lets your code detect when the user is looking at an object that should provide an action.

With Raycast setup on the camera, you need to update the TravelerTemplate to know how to react to an intersection with the Raycast. To do this, add the VRInteractiveItem script to the TravelerTemplate. This script is a quick setup for working with events like when the Raycast is over an object, or when the Raycast leaves an object, or even capturing the click event.

Next, you create a script called TravelerInteraction, which implements the events in the VRInteractiveItem script. For this app, you'll use the OnOver, OnOut and OnClick events. When the user looks at a Traveler object, the Traveler will change in appearance. When the user looks away, the Traveler object will change back.

When the user is looking at the Traveler object and clicks, a dialog will appear showing the traveler's name, title and destination. Here's the TravelerInteraction script:

```
public class TravelerInteraction : MonoBehaviour{
    public VRInteractiveItem InteractiveItem;
    public Shader Highlight;
    public TravelerManager TravelerManager;
    public Canvas Dialog;
    private Shader standardShader;
    private bool isDialogShowing;
}
```

The VRInteractiveItem is needed to work with the VREyeRaycast and assign methods to the VRInteractiveItem events.

The VRInteractiveItem is needed to work with the VREyeRaycast and assign methods to the VRInteractiveItem events. Highlight is a custom shader that'll be used to highlight the Traveler object when the user looks at the object. I won't go into custom shaders here as they're a complex topic. To learn more, see the documentation on the Unity site at bit.ly/2agfb7q. TravelerManager is a reference to the TravelerManager script, which will be used to look up details about the selected Traveler from the TravelerManager.TravelerList collection. The final public property here is the Canvas game object that will be used to show details about the traveler when clicked. Private variables include a reference to the standard shader to return the Traveler object to after you stop looking at the traveler and a Boolean value to determine if the dialog is showing.

To get started, add the Enabled method to the TravelerInteraction class to connect the events of the VRInteractiveItem class to methods within the TravelerInteraction class:

```
private void OnEnable(){
    InteractiveItem.OnOver += HandleOver;
    InteractiveItem.OnOut += HandleOut;
    InteractiveItem.OnClick += HandleClick;
}
```

HandleOver uses the Renderer component to change the shader to a custom shader. The Highlight property lets a designer use the Unity Editor to set the Shader for highlighting the Traveler.

Figure 3 Display Dialog When Object Is Clicked

```
private void HandleClick(){
    if (!isDialogShowing){
        var textComponent =
            Dialog.gameObject
            .GetComponentInChildren<Text>();
        var travelerInfo =
            TravelerManager.TravelerList.SingleOrDefault(x => x.Name ==
            this.gameObject.name);
        textComponent.text = String.Format("{0}\n{1}",
            travelerInfo.Name, travelerInfo.DestCity);
        Dialog.gameObject.SetActive(true);
        isDialogShowing = true;}
    else{
        Dialog.gameObject.SetActive(false);
        isDialogShowing = false;}
}
```

```
private void HandleOver(){
    var renderer =
        gameObject.GetComponentInChildren<Renderer>();
    renderer.material.shader = Highlight;
}
private void HandleOut(){
    var renderer =
        gameObject.GetComponentInChildren<Renderer>();
    renderer.material.shader = standardShader;
}
```

When the user looks at a Traveler, the Highlight shader is applied to differentiate the Traveler from the other travelers. This provides a visual queue to the user and whichever Traveler currently has the user's gaze. When the user looks away the HandleOut method will be triggered and return the Traveler to the standardShader value.

Now, you want to show the traveler information to users when they click on the traveler. To do this, first you need to add System.Linq to the using statements for the TravelerInteractive class so that you can find the traveler by the game object's name. Also, you need to add the UnityEngine.UI namespace to work with the Text component in the dialog, as shown in **Figure 3**.

In this code block, first you determine if the dialog is displayed or not. If it is displayed, then disable the dialog to hide it from view. If the dialog isn't showing, then get the text object and set its value to the name and destination of the Travelers. When you originally created the traveler object, you used the traveler name as the name of the game object used to represent the traveler. Using LINQ, you can search the traveler list by that name to return the traveler's detail information. For simplicity sake you use the traveler's name but in larger, more complex systems, a unique identifier would be most ideal to use so that each traveler is unique. A quick note on LINQ, remember that performance is key in building a mobile VR experience. LINQ can do great things—but it does have overhead—so be aware of where LINQ is used and determine if it's needed or just a convenience. Here LINQ is used for brevity. Once the text is set to the name and destination city of the traveler, the dialog is enabled (which is a canvas object) to display the text to the user.

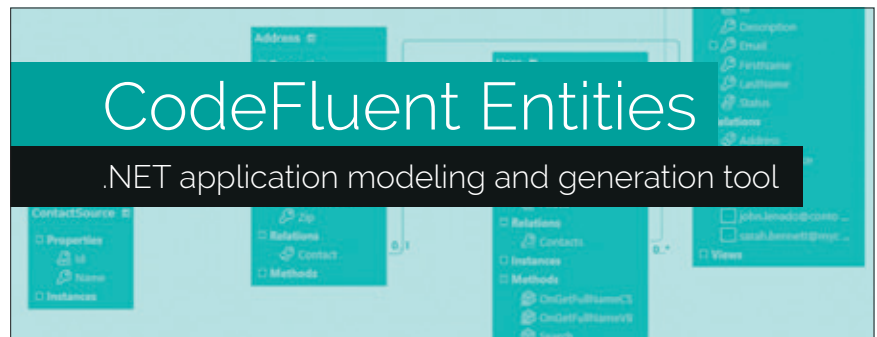
Wrapping Up

In this article, I reviewed the basics of building a UI, getting data from a service and interacting with objects in a VR world. Using these skills, the world of ContosoTravel can grow into a rich app with search, perhaps more levels to represent regions of the world and

animations to bring the avatars to life (such as wandering around). Explore these fundamentals to build VR worlds for today and prepare for the mixed reality users of tomorrow. ■

TIM KULP is the principle engineer at bwell in Baltimore, Md. He's a Web, mobile and UWP app developer, as well as author, painter, dad and "wannabe Mad Scientist Maker." Find him on Twitter: @seccode or via LinkedIn: linkedin.com/in/timkulp.

THANKS to the following Microsoft technical expert for reviewing this article: Adam Tuliper



Create **high-quality software**
always aligned with your business rules!

Ultimate Edition now only USD699 / year
Free edition still available

If you like to start from a visual model rather than dive in code first, if you like to focus on business code rather than continuously write plumbing code, give it a try.

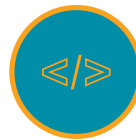
Code and model are kept in sync and leverage best practices to ensure application maintainability and performance as well as developer productivity.

CodeFluent Entities is fully integrated in Visual Studio 2008 to 2015.



MODEL YOUR BUSINESS

Leverage the graphical modeler to define your business model in a centralized place



DEFINE YOUR FOUNDATIONS

Generate a direct translation of your model into ready-to-use data and business layers



DEVELOP YOUR APPLICATIONS

Build modern applications by focusing on your high-value custom code

soft<luent

Innovative software company founded in 2005 by Microsoft veterans

SoftFluent - 2018 156th Avenue NE Bellevue, WA 98007 USA - info@softfluent.com - www.softfluent.com

Copyright © 2005 - 2016, SoftFluent SAS. All rights reserved.

Microsoft, Visual Studio and Excel® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.



The Source of Truth: The Role of Repositories in DevOps

Kraig Brockschmidt

The first article of this series, “From Code to Customer: Exploring Mobile DevOps” (msdn.com/magazine/mt767694), introduced the Microsoft DevOps stack for mobile apps and back-end services. It outlined the stages of the whole *release pipeline*, shown in **Figure 1**. A release pipeline, put succinctly, is how code that’s committed to a source repository gets transformed into customer-ready apps and services, and then delivered to customer devices and customer-accessible servers. A release pipeline, at its core, is simply a list of the steps that are necessary to make a release happen. Indeed, the practice of DevOps begins with being clear about the exact steps and the processes involved in a release. From there it’s relatively easy to incrementally automate those processes using tooling within the Microsoft DevOps stack.

This article discusses:

- The motivation behind source control systems, which are a form of automation
- Options for source control with Visual Studio Team Services
- Communication and auditing capabilities within source control systems
- Team Projects in Visual Studio Team Services
- Creating and populating repositories in Team Services and Visual Studio

Technologies discussed:

Visual Studio, Visual Studio Team Services, Team Foundation Server, Git, GitHub, Team Foundation Version Control

As discussed in the first article, you should always understand how to do every step in a release pipeline manually. Many app projects get started with manual processes, such as manual builds, to get feedback from testers as early as possible. Knowing how to do everything manually also provides a fallback in case some part of an automated release pipeline breaks down.

However, manual processes are expensive to scale, prone to human error, often tedious and put every step at risk of competing for the attention of your employees with all their other work. Automation—having a computer perform those tasks—provides much better scaling, reliability, predictability and auditing, which means higher quality at a lower cost. Automation also frees your employees for tasks that really do need human attention.

In this article I’ll explore a very important aspect of the release pipeline, one that’s probably taken for granted: source control. A project’s source code is the *input* to the release pipeline, as outlined in **Figure 1**, and most developers today accept managing code in a source control system as a matter of course. But it helps you understand the whole of DevOps better if you can clearly see the essential role source control plays in that context.

The Reasons for Source Control

The first step in automating a release pipeline is to manage your code in a source control repository of some kind. Again, the source code is the input to the release pipeline, and the next stage, Build, is what converts that code into the artifacts that are then fed into the rest of the pipeline. To automate the build process, with continuous integration especially, systems like Visual Studio Team Services

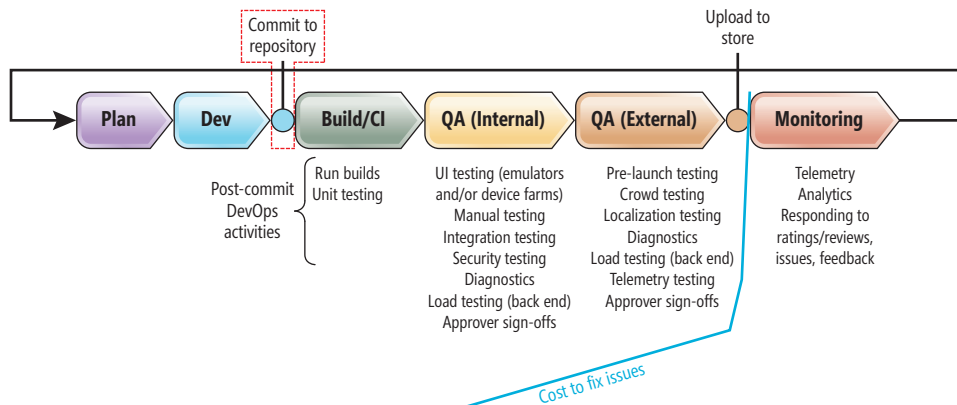


Figure 1 The Source Control Repository Is the Input to the Release Pipeline

(Team Services for short) must be able to detect when changes are made to that repository. As such, it's not sufficient to manage your source code merely in some arbitrary folder on your hard drive.

Note: If you don't already have a free Team Services account, create one by following the instructions at bit.ly/29xK30o. Better still, check out the Visual Studio Dev Essentials program (bit.ly/29xKCqy), which gives you a Team Services account along with easy access to many other services, including \$25 in Azure credit for 12 months.

I'll talk about build and continuous integration in the next article of the series. Here I want to focus specifically on source control itself, starting with a brief review of *why* source control exists in the first place, and the general role it plays in DevOps as a whole. My reason for this is to point out something you might never have thought about before: Source control is fundamentally a form of automation.

This statement may surprise you, because most developers today consider source control a given. But that wasn't always the case. Chances are you've worked on projects without source control at some point in your career, especially when you were first getting started. In those projects, you likely just had a folder on your local hard drive containing all your code, which is what you get when you create a new local app project in Visual Studio. From there, you may have run a local build to produce the executables and such needed for distribution, which you then manually uploaded to a public Web server or perhaps shared with others on physical media.

In short, source control is in no way required to produce a customer-ready app and its back-end services. But having only a single local copy of your source code has a number of problems and risks, all of which I imagine you've experienced directly:

- If your hard drive crashes, you might lose everything.
- A single copy of the source code doesn't maintain any change history, making it very difficult to revert to a prior working state of the project.
- Multiple people working on the code can easily overwrite each other's changes, or introduce breaking changes, without anyone knowing.

It's certainly possible to mitigate these risks manually, to some extent. Regular backups, for instance, help guard against code loss and provide a certain degree of history. However, the nongranular nature of whole-project backups makes it very difficult to revert only

parts of the project while leaving other changes intact. It's also possible for people on a team to make personal copies of the code to avoid conflicts, but it's then exceptionally tedious to integrate those copies together into a working state. Team members can also communicate breaking changes to others by direct e-mail or other messaging, but this becomes burdensome to do on a consistent basis.

Of course, as developers we generally avoid such burdens as often as we can. Instead, we find creative ways to automate these tasks, which is exactly what source control is all about.

At a high level, source control means the following:

- Maintaining a shared repository of all project code on a server, with some kind of automatic backup mechanism.
- Logging changes on a per-file basis, so you can see the entire history for any given file, as well as changes to multiple files that were committed to the repository as a group. This makes it easy to associate build failures and test regressions with specific changes, and to revert individual files or groups of files to any previous state, not just the state of the last backup.
- Storing the code in a place where a build system can detect changes to the repository and automatically trigger a build, which means performing an immediate integration test (continuous integration) with those changes.
- Managing overwrites and conflicts among multiple users, either by requiring developers to lock files for exclusive access while they're working on them, or by having tools that can automatically merge changes and detect conflicts.
- Sending automatic notifications to interested developers when certain files change, or when merge conflicts require manual resolution.

In short, source control automates many of the tedious processes involved in maintaining a dependable and auditable repository of a project's code. It's essential for managing project code for both single developers and teams alike, and is the basis for an automated release pipeline.

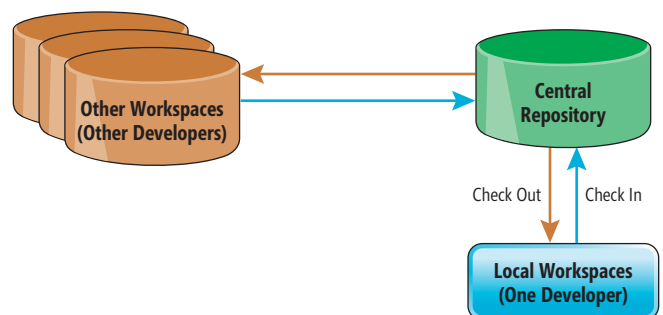


Figure 2 The Basic Team Foundation Version Control Relationships

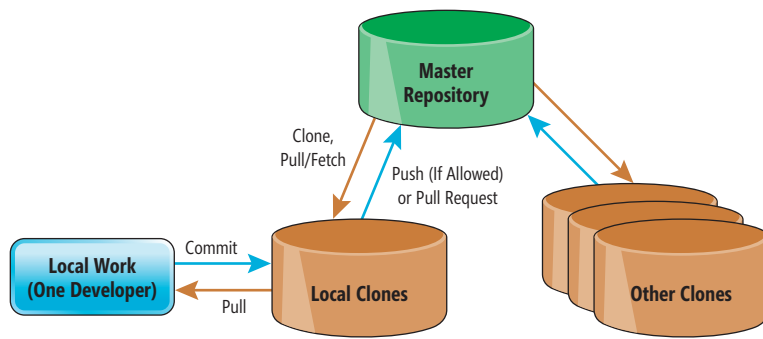


Figure 3 The Basic Git Relationships

Source Control Options

Given the pervasive need for source control, it's no surprise that many different systems have evolved over the years. Here, though, I'll discuss the two you can host directly within Team Services: Git and Team Foundation Version Control (TFVC). Hosting a repository means that it's directly stored and managed within your Team Services account. The Team Services build system can also draw from external Git, GitHub and Subversion repositories, as well, which I'll talk about in the next article in the series.

Which source control system you choose for a project is really a matter of preference, the experience of your development team and cost considerations. GitHub, for example, is free for public repositories, but has a cost for private ones (see github.com/pricing). Public GitHub repositories are open to everyone by default, which is why open source projects are typically hosted on GitHub. Many of the documentation sets I work on at Microsoft, for example, are stored there, including the entire collection of Microsoft Azure documentation (github.com/Azure/azure-content), and the docs for the Visual Studio Tools for Apache Cordova (github.com/Microsoft/cordova-docs). I also use GitHub for a variety of individual sample projects, like

the Altostratus sample (github.com/kraigb/Altostratus) that appeared in *MSDN Magazine* last year (bit.ly/29mKHIC). Similarly, I chose GitHub for the MyDriving project (github.com/Azure-Samples/MyDriving) because it was intended to be open source from the beginning. (For more on MyDriving overall, see aka.ms/iotsampleapp.)

Team Services, on the other hand, is oriented toward private repositories (Git or TFVC) by default, which means that when you first create a repository, only you have access. To give access to others, you must specifically add them as team members (see bit.ly/29QDHqI). The advantage is that you can host unlimited private repositories within your Team Services account for free

for as many MSDN subscribers as you want, and costs only kick in when you have more than five users without MSDN subscriptions. For these reasons, I use Team Services for personal projects, such as apps I have in app stores and for code I want to share with only specific individuals.

Which source control system you choose for a project is really a matter of preference, the experience of your development team and cost considerations.

The primary functional difference between Git and TFVC lies in their respective source control models. A full comparison can

be found in the documentation in the topic "Choosing the right version control for your project" (bit.ly/29oZKTZ), but let me summarize.

TFVC, illustrated in **Figure 2**, is *centralized*, meaning that files live in a single, central, read-only repository, for which the administrator is responsible for backups. To do work with TFVC, you typically maintain a local read-only copy of the latest version of files from the repository (called a workspace), from which you can run builds and test the app. To make changes, you check out one or more files, which gives you exclusive access until those files are checked back in (that is, integrated into the repository). TFVC then merges the changes into the repository. If multiple developers check out the same file simultaneously (which is allowed), TFVC detects merge conflicts on check-in and informs the developers if manual resolution is needed.

Git is *distributed*, meaning that although the master repository lives on the host, you "clone"

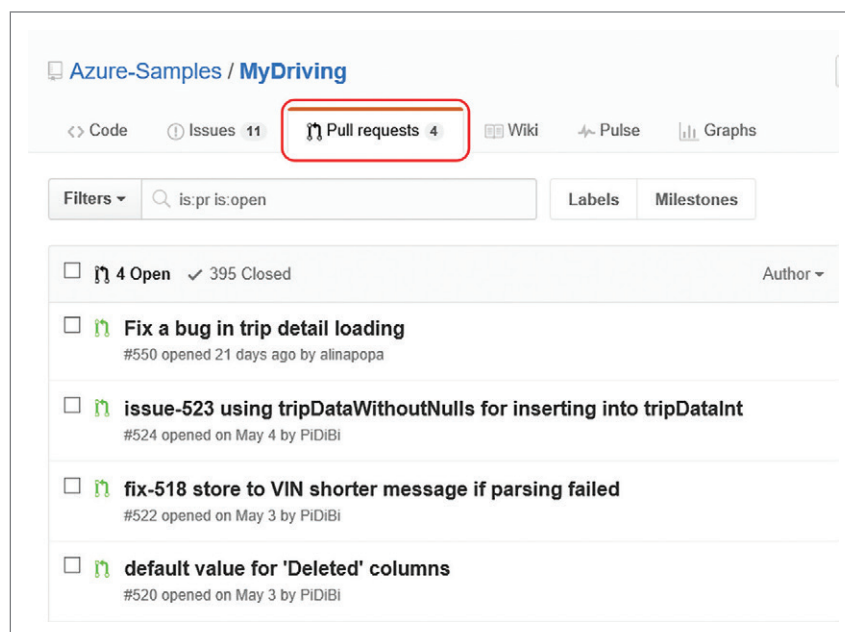


Figure 4 Pull Requests List on GitHub for the MyDriving Project

We didn't invent the Internet...

...but our components help you power the apps that bring it to business.



TOOLS • COMPONENTS • ENTERPRISE ADAPTERS

- **E-Business**
AS2, EDI/X12, NAESB, OFTP ...
- **Credit Card Processing**
Authorize.Net, TSYS, FDMS ...
- **Shipping & Tracking**
FedEx, UPS, USPS ...
- **Accounting & Banking**
QuickBooks, OFX ...
- **Internet Business**
Amazon, eBay, PayPal ...
- **Internet Protocols**
FTP, SMTP, IMAP, POP, WebDav ...
- **Secure Connectivity**
SSH, SFTP, SSL, Certificates ...
- **Secure Email**
S/MIME, OpenPGP ...
- **Network Management**
SNMP, MIB, LDAP, Monitoring ...
- **Compression & Encryption**
Zip, Gzip, Jar, AES ...



The Market Leader in Internet Communications, Security, & E-Business Components

Each day, as you click around the Web or use any connected application, chances are that directly or indirectly some bits are flowing through applications that use our components, on a server, on a device, or right on your desktop. It's your code and our code working together to move data, information, and business. We give you the most robust suite of components for adding Internet Communications, Security, and E-Business Connectivity to

any application, on any platform, anywhere, and you do the rest. Since 1994, we have had one goal: to provide the very best connectivity solutions for our professional developer customers. With more than 100,000 developers worldwide using our software and millions of installations in almost every Fortune 500 and Global 2000 company, our business is to connect business, one application at a time.

connectivity
powered by 

To learn more please visit our website →

www.nsoftware.com

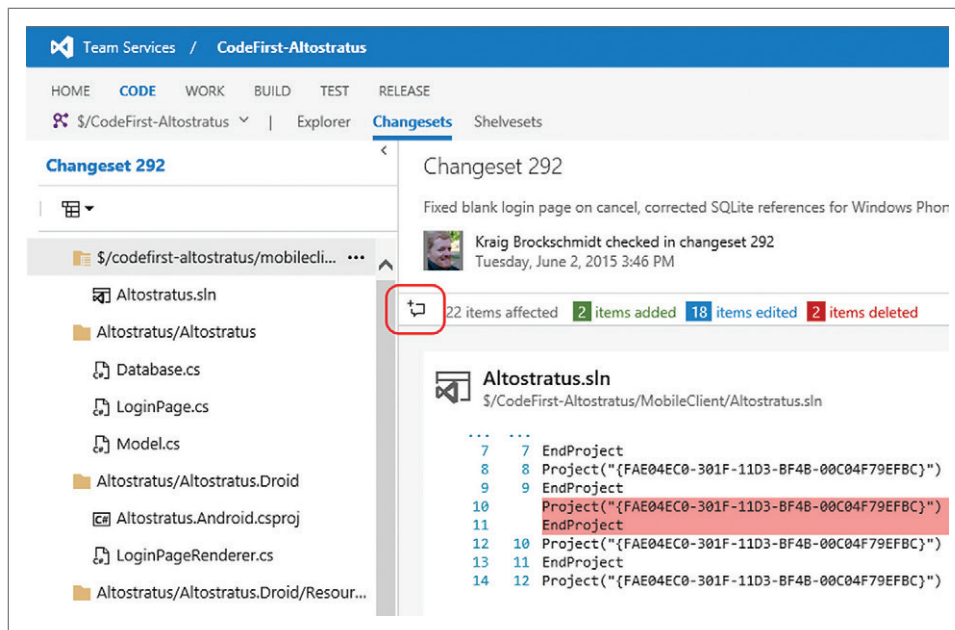


Figure 5 The UI for a Changeset in Visual Studio Team Services, Showing the Commenting Button

the entire repository (change history and all), do your work locally and then commit completed changes to the clone as illustrated in **Figure 3** (see also git-scm.com for details on Git workflows). When ready to integrate changes with everyone else's work, you submit "pull requests" from your clone into the master. In this model, every clone is effectively a backup of the repository.

Note that both Git and TFVC use some similar words like "check out" to describe entirely different processes, which can be confusing. It's best to simply work with each one by itself and not expect to transfer any knowledge between the two.

Git's pull request mechanism is able to detect when changes can be merged automatically, and when manual resolution is needed. Of course, in public repositories like GitHub, you don't necessarily want anyone and everyone to be able to merge pull requests. Typically, only certain individuals will have permission to merge pull requests, who then act as the gatekeepers for the integrity of the repository as a whole.

For example, take a look at the top of the MyDriving repository at github.com/Azure-Samples/MyDriving and you'll see Pull requests (see **Figure 4**). Clicking on that shows currently open requests that are waiting for moderation by those with merge permissions. You can also look at the whole history of pull requests by clicking on the Closed tab in the list.

Both TFVC and Git support what's known as branching, which essentially means creating another layer between the master or central repository and other clones or copies. This allows sub-groups of developers (and testers) to do significant work in that branch without affecting the master repository. A branch also maintains its own change history and, in the case of Git, manages its own pull requests from each clone. Only

when that team is ready to integrate the work in the branch with the master do they submit a pull request into the master. For more, see "Use branches to isolate risk in TFVC" (bit.ly/29ndIQz) and "Create work in branches" (bit.ly/29WmgY) in the documentation.

Communication and Auditing

In both Git and TFVC—and in source control systems generally—check-ins, pull requests and so forth all have an associated commenting mechanism. That's how you communicate the changes you've made to everyone else working on the project, and how teams can discuss those changes. These systems typically also provide notifications and discussions

around broader issues (like open bugs), work items and so forth.

On GitHub, for example, you include comments with each code commit to your clone, and then make additional comments when submitting a pull request. Those responsible for checking and merging that request can then leave comments or questions within the pull request, especially if they find potential problems with the new code. You can see plenty of examples if you click around in the Closed pull request list in the MyDriving repository mentioned earlier. Similarly, click on the Issues tab on the homepage to see more discussions. As for notifications, these are managed through your personal settings in the Notifications section.

Team Services, for its part, has a whole system for tracking work items, bugs and so on, which you can read about in the Agile tools section of the Team Services documentation (bit.ly/29tvKIE). Within a code repository (whether Git or TFVC), there's a ubiquitous commenting control, as shown in **Figure 5**, that lets team members leave notes on changesets (groups of changes), individual files and so on.

Such comments, along with specific details about what changes were made to what files, all go into the repository's history or change log. Having an extensive, detailed history of who made what

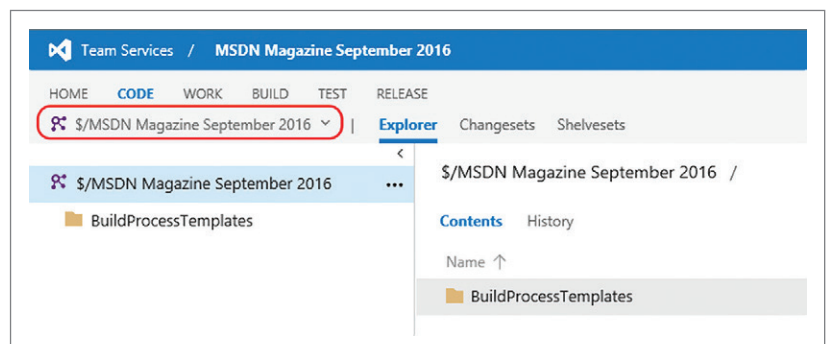


Figure 6 The Code Tab for a New Project Using Team Foundation Version Control

Visit Us at
Microsoft Ignite
Booth #470



Get Connected Today.
TAKE A FREE TEST DRIVE!
www.MelissaData.com/kyc

Know Your Customer

Customers share tons of personal data through an increasing number of channels and applications. We supply the full spectrum of data quality solutions to solve your KYC challenges – to collect, verify, enrich and consolidate clean contact data for a true and complete view of the customer. Available in Microsoft®, Oracle®, Pentaho®, Salesforce® and more.

Solutions for 240+ Countries



10,000+ Customers Worldwide



30+ Years Strong



Data Quality & Mailing Solutions



Cloud • On-Premise • Services



Germany
www.MelissaData.de

United Kingdom
www.MelissaData.co.uk

India
www.MelissaData.in

Australia
www.MelissaData.com.au

MELISSA DATA®

Your Partner in Global Data Quality

www.MelissaData.com | 1-800-MELISSA

changes at what time—along with any discussion that happened around those changes—is one of the most significant benefits of using a source control system. The history makes the entire repository *auditable* across time. If unexpected problems come up later on in the release pipeline, such as regressions that are revealed through unit, integration or UI testing, it's easy to go back and see the specific changes in a particular file that caused that regression.

Making that process “easy” is actually very important where DevOps as a whole is concerned. Recall from the first article in this series that I talked about DevOps as the continuous validation of performance for an app and services, where “performance” includes both the customer experience and the cost of production. The ability to discover defects as early as possible in the release pipeline helps to minimize costs. Equally important is the time it takes to pinpoint *where*, exactly, a defect actually exists—the quicker the better! Indeed, you've probably had the experience of spending many frustrating days trying to track down a bug in some project, only to find that the fix took all of 10 seconds. In such cases, nearly all the cost came from merely locating the bug.

This is a very important consideration if you or anyone on your team objects to using source control, rather than just “winging it” by keeping your code on some simple network share. The little bit of investment you make by adopting a source control system pays huge dividends over the lifetime of the project, especially when combined with automated builds, continuous integration and automated testing, as you'll see in future articles. Continuous integration means that every code change triggers an automatic build and runs automated tests, so if that code change causes any kind of failure (build or test), you know about it within minutes.

Team Projects and Multiple Repositories

To build an automated release pipeline for your app and services within Team Services or Team Foundation Server (TFS), you begin with what's called a team project. A team project is the container for everything you do in Team Services, including planning, work item tracking, collaboration via team rooms, builds, continuous integration, test management, release management, and source control repositories. I say “repositories” here because a team project can directly host multiple repositories, and the Team Services build system can also draw from external Git, GitHub and Subversion repositories. (Similarly, TFS can draw from

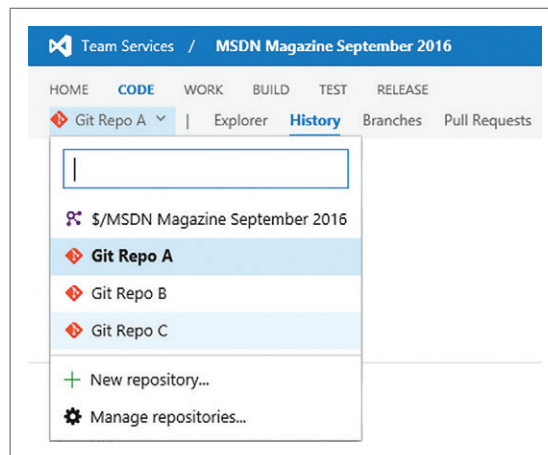


Figure 7 A Team Project with Multiple Git Repositories and One Team Foundation Version Control Repository

dialog box in which you select either Git or TFVC as the source control model for the project's default repository. This is really just for convenience. If you select TFVC you can create Git repositories later, as you'll see shortly; if you select Git, you can add a TFVC repository later along with more Git repositories. And if you prefer to use an external host like GitHub, you won't use the default repository at all and this choice is irrelevant. For example, when I set up a team project for MyDriving, I selected Git for the default repository, but all that's there is a default readme file because the real project is hosted entirely on GitHub.

To show how a single team project can manage multiple repositories, I created an example project in my own Team Services account and selected TFVC. This project's Code tab initially appears as shown in Figure 6. Clicking on the outlined control shows the list of existing repositories in the Team Project, along with the New repository and Manage repositories commands. The New command brings up a dialog box where you again select between Git and TFVC. Because I initially created the team project with TFVC, I can't create another TFVC repository (only one is allowed), but

I can create any number of additional Git repositories, as shown in Figure 7.

The Manage repositories command takes you to the Team Services control panel where you can see all the repositories (and Git branches) at once; rename or delete them; and manage access permissions. For all the details around users, groups and permissions—too many to cover here—refer to the documentation for “Permissions and groups in Team Services” (bit.ly/29nxvpd) in the “Git repository” and “TFVC” sections. Suffice it to say you can exercise a very fine-grained control over permissions for every one of your team members. Of course, if you're using an external source control system like GitHub, you'll manage permissions on that

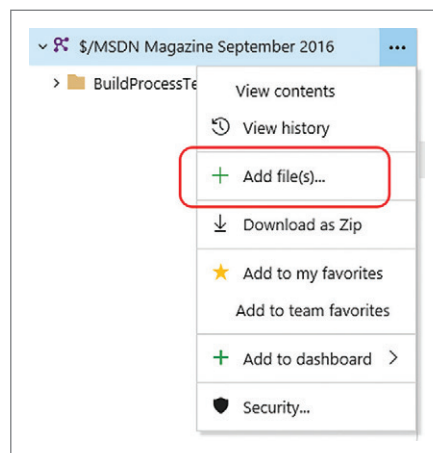


Figure 8 Visual Studio Team Services Command to Add Files to a Team Foundation Version Control Repository

BUSINESS FILE APIS

TRY RISK FREE - 30 Day Trial

Open, Create, Convert, Print and Save files from your apps!



Aspose.Total

Aspose.Cells

XLS, CSV, PDF, SVG, HTML, PNG
BMP, XPS, JPG, SpreadsheetML...

Aspose.Words

DOC, RTF, PDF, HTML, PNG
ePUB, XML, XPS, JPG...

Aspose.Pdf

PDF, XML, XSL-FO, HTML, BMP
JPG, PNG, ePUB...

Aspose.Slides

PPT, POT, ODP, XPS, HTML
PNG, PDF...

Aspose.BarCode

JPG, PNG, BMP, GIF, TIF, WMF
ICON...

Aspose.Tasks

XML, MPP, SVG, PDF, TIFF
PNG...

Aspose.Email

MSG, EML, PST, MHT
OST, OFT...

Aspose.Imaging

PDF, BMP, JPG, GIF, TIFF
PNG...

+ more!

.NET

Java

Cloud

Download FREE trial at www.aspose.com.

Contact Us:

US: +1 903 306 1676

EU: +44 141 628 8900

AU: +61 2 8006 6987

sales@aspose.com

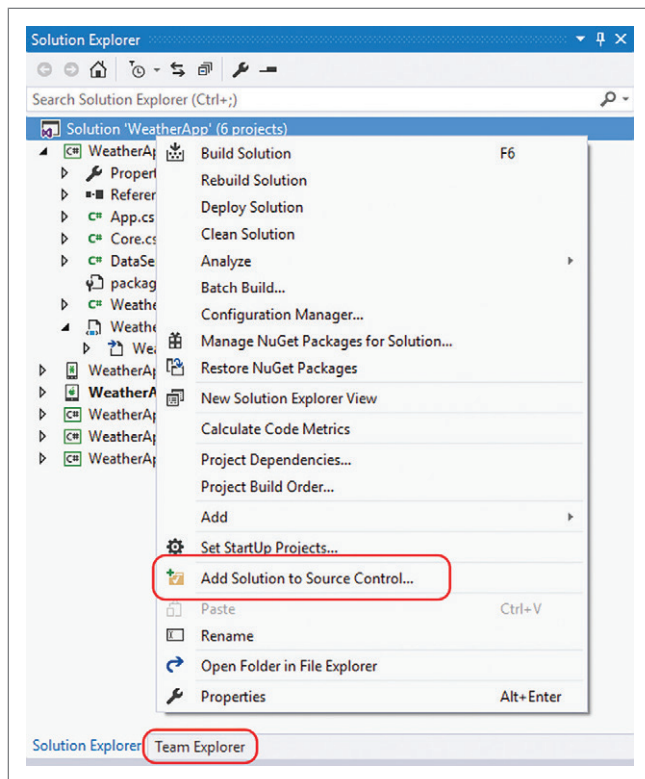


Figure 9 Adding a Solution to Source Control in Visual Studio

site instead. Note, however, that permissions for the team project as a whole—and not the repositories—are managed through the Security tab shown in that UI. You'll also find those details in the same documentation page noted earlier (bit.ly/29nxvpd).

Populating a Team Services Repository with Code

Once you've created a repository, the big question is how you get your code into it. Team Services gives you a variety of means, as does Visual Studio, so to close this article let me briefly run through those options.

For a TFVC repository, you can first upload files into the repository or create new files directly through the Team Services portal. Navigate first to the Code tab in the team project, then click the ... next to the repository and select + Add File(s), as shown in **Figure 8**. This brings up a dialog box (not shown) through which you can create files and edit them directly in the portal, or create a list of files to upload. In the latter case you also include a check-in comment.

The other way to add code to a TFVC repository is through the Visual Studio Solution Explorer. This is a very convenient way to take a local solution and transfer all the code into source control. Just right-click the solution and select Add Solution to Source Control, as shown in **Figure 9**. This will bring up a dialog box in which you can select the appropriate team project within your Team Services account or on a specific TFS machine. To connect to a server, which you may need to do initially, use Team Explorer in Visual Studio (the tab outlined at the bottom of **Figure 9**). For details, see the "Work in Team Explorer" topic in the Visual Studio documentation (bit.ly/29oGp5j).

For Git repositories, Team Services will automatically prompt you with a variety of options when you create the repository or navigate

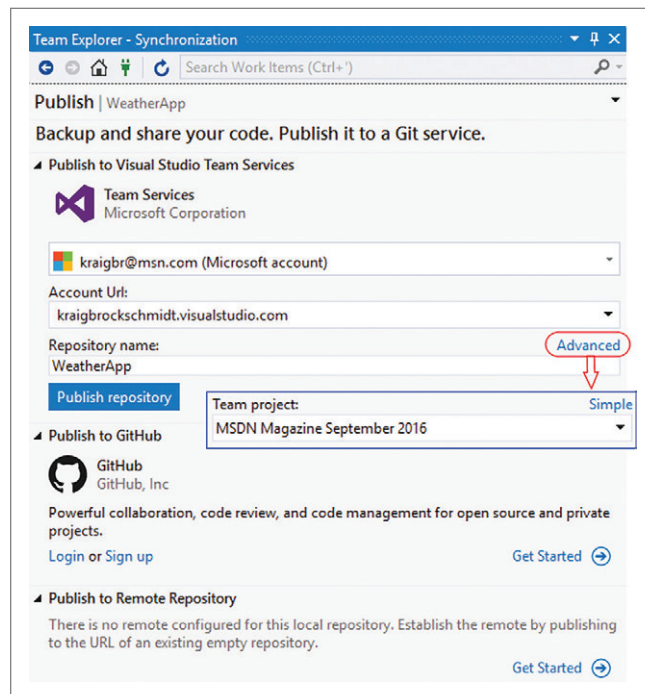


Figure 10 The Publish UI in Visual Studio

to it on the Code tab, a UI that speaks sufficiently for itself. Alternatively, you can create a local Git repository in Visual Studio, and then publish it to a repository in Team Services. I won't go through that process in detail here, because it's nicely documented in the topic "Share Your Code with Git and Visual Studio" (bit.ly/29VDYJg). The short of it, though, is to click the Publish button on the lower right of the Visual Studio status bar, which brings up the Publish tab of Team Explorer as shown in **Figure 10**. Here you can select the Team Services account to use. An important detail is clicking on the small Advanced text to select a team project as shown. As you can also see, the same UI gives you an easy path to publish code to GitHub and other external repositories, as well.

Looking Ahead

With a source control repository in place, you're now ready to take the next step in automating your release pipeline by setting up builds and continuous integration. That's what I'll cover in my next article, where you'll see how any given build definition within Visual Studio Team Services can draw from any of the source control repositories I've discussed here. A team project, furthermore, can manage any number of such build definitions, which means the team project can coordinate builds from any number of repositories to produce the necessary artifacts for the rest of the release pipeline, or *pipelines*, as the case may be. ■

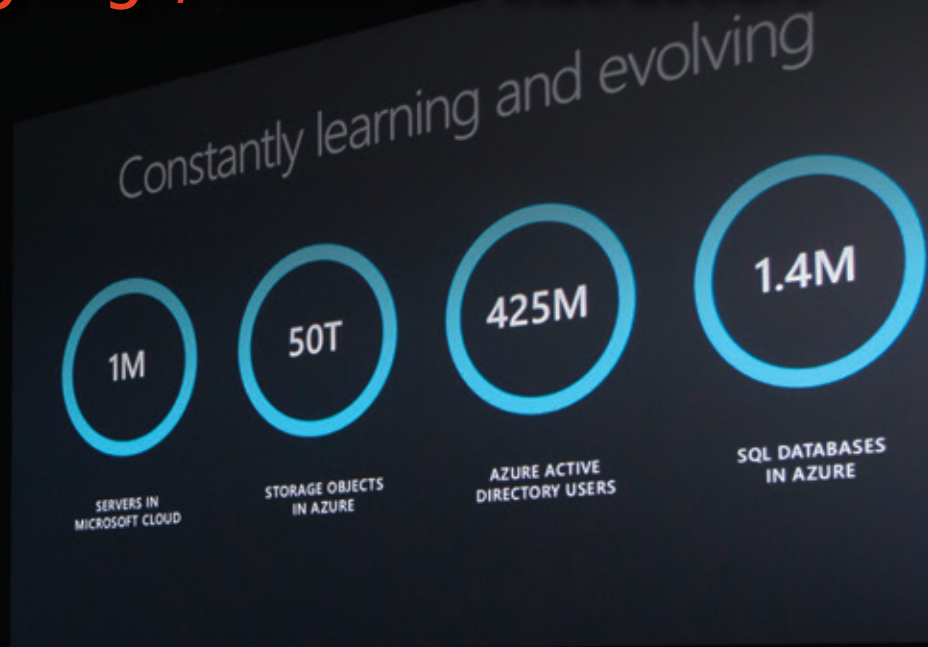
KRAIG BROCKSCHMIDT works as a senior content developer for Microsoft and is focused on DevOps for mobile apps. He's the author of "Programming Windows Store Apps with HTML, CSS and JavaScript" (two editions) from Microsoft Press and blogs on kraigbrockschmidt.com.

THANKS to the following technical experts for reviewing this article:
Donovan Brown and Gordon Hogenson

Learn By Doing!

Stay Competitive in a Cloud-First World with Opsgility

Visit us @ Ignite – Booth #134 to learn more about our cutting edge, hands-on instruction.



opsgility

Opsgility Learning Paths



REAL CODE. REAL LABS. REAL LEARNING.

[Opsgility.com](https://opsgility.com)

Feature Slices for ASP.NET Core MVC

Steve Smith

Large Web apps require better organization than small ones. With large apps, the default organizational structure used by ASP.NET MVC (and Core MVC) starts to work against you. You can use two simple techniques to update your organizational approach and keep up with a growing application.

The Model-View-Controller (MVC) pattern is mature, even in the Microsoft ASP.NET space. The first version of ASP.NET MVC shipped in 2009 and the first full reboot of the platform, ASP.NET Core MVC, shipped early this summer. Throughout this time, as ASP.NET MVC has evolved, the default project structure has remained unchanged: folders for Controllers and Views and often for Models (or perhaps ViewModels). In fact, if you create a new ASP.NET Core app today, you'll see these folders created by the default template, as shown in **Figure 1**.

There are many advantages to this organizational structure. It's familiar; if you've worked on an ASP.NET MVC project in the last few years, you'll immediately recognize it. It's organized; if you're looking for a controller or a view, you have a good idea where to start. When you're beginning a new project, this organizational structure works reasonably well, because there aren't yet many

files. As the project grows, however, so does the friction involved in locating the desired controller or view file within the growing numbers of files and folders in these hierarchies.

To see what I mean, imagine if you organized your computer files in this same structure. Instead of having separate folders for different projects or kinds of work, you had only directories organized solely by what kinds of files. There might be folders for Text Documents, PDFs, Images and Spreadsheets. When working on a particular task that involves multiple document types, you would need to keep bouncing between the different folders and scrolling or searching through the many files in each folder that are unrelated to the task at hand. This is exactly how you work on features within an MVC app organized in the default fashion.

The reason this is an issue is that groups of files organized by type, rather than purpose, tend to lack *cohesion*. Cohesion refers to the degree to which elements of one module belong together. In a typical ASP.NET MVC project, a given controller will refer to one or more related views (in a folder corresponding to the controller's name). Both the controller and the view will reference one or more ViewModels related to the controller's responsibility. Typically, though, few ViewModel types or views are used by more than one controller type (and, typically, the domain model or persistence model is moved to its own separate project).

A Sample Project

Consider a simple project tasked with managing four loosely related application concepts: Ninjas, Plants, Pirates and Zombies. The actual sample only lets you list, view and add these concepts. However, imagine there's additional complexity that would involve more views. The default organizational structure for this project would look something like **Figure 2**.

This article discusses:

- ASP.NET Core MVC
- ASP.NET Core MVC Areas
- Feature Folders in ASP.NET Core MVC

Technologies discussed:

ASP.NET Core MVC

Code download available at:

bit.ly/29Mxsl0

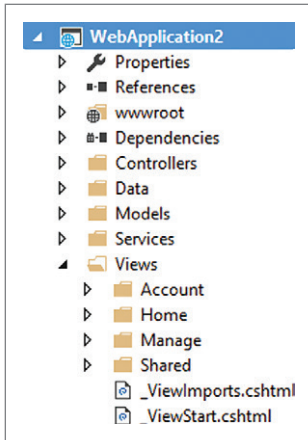


Figure 1 Default ASP.NET Core Web App Template Structure

To work on a new bit of functionality involving Pirates, you would need to navigate down into Controllers and find the PiratesController, and then navigate down from Views into Pirates into the appropriate view file. Even with only five controllers, you can see that's a lot of folder up-and-down navigation. This is often made worse when the root of the project includes many more folders, because Controllers and Views aren't near one another alphabetically (so additional folders tend to fall between these two in the folder list).

An alternative approach to organizing files by their type is to organize them along the lines of what the application *does*. Instead of folders for Controllers, Models, and Views, your project would have folders organized around features or areas of responsibility. When working on a bug or feature related to a particular feature of the app, you would need to keep fewer folders open because the related files could be stored together. This can be done in a number of ways, including using the built-in Areas feature and rolling your own convention for feature folders.

How ASP.NET Core MVC Sees Files

It's worth spending a moment to talk about how ASP.NET Core MVC works with the standard kinds of files an application built in it uses. Most of the files involved in the server side of the application are going to be classes written in some .NET language. These code files can live anywhere on disk, as long as they can be compiled and referenced by the application. In particular, Controller class files don't need to be stored in any particular folder. Various kinds of model classes (domain model, view model, persistence model and so on) are the same, and can easily live in separate projects from the ASP.NET MVC Core project. You can arrange and rearrange most of the code files in the application in whatever way you like.

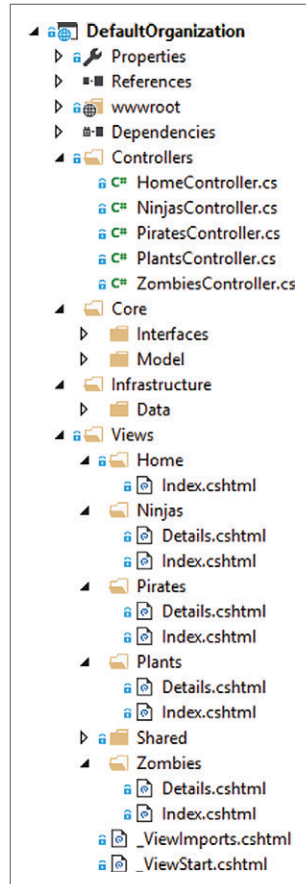


Figure 2 Sample Project with Default Organization

Views, however, are different. Views are content files. Where they're stored relative to the application's controller classes is irrelevant, but it's important that MVC knows where to look for them. Areas provide built-in support for locating views in different locations than the default Views folder. You can also customize how MVC determines the location of views.

Organizing MVC Projects Using Areas

Areas provide a way of organizing independent modules within an ASP.NET MVC application. Each Area has a folder structure that mimics the project root conventions. Therefore, your MVC application would have the same root folder conventions, and an additional folder called Areas, within which would be one folder for each section of the app, containing folders for Controllers and Views (and perhaps Models or ViewModels, if desired).

Areas are a powerful feature that let you segment a large application into separate, logically distinct sub-applications. Controllers, for example, can have the same name across areas, and in fact, it's common to have a HomeController class in each area within an application.

To add support for Areas to an ASP.NET MVC Core project, you just need to create a new root-level folder called Areas. In this folder, create a new folder for each part of your application you want to organize within an Area. Then, inside this folder, add new folders for Controllers and Views.

Your controller files should thus be located in:

```
/Areas/[area name]/Controllers/[controller name].cs
```

Your controllers need to have an Area attribute applied to them to let the framework know they belong within a particular area:

```
namespace WithAreas.Areas.Ninjas.Controllers
{
    [Area("Ninjas")]
    public class HomeController : Controller
    {
    }
}
```

Your views should then be located in:

```
/Areas/[area name]/Views/[controller name]/[action name].cshtml
```

Any links you had to views that have moved into areas should be updated. If you're using tag helpers, you can specify the area name as part of the tag helper. For example:

```
<a asp-area="Ninjas" asp-controller="Home" asp-action="Index">Ninjas</a>
```

Links between views within the same area can omit the asp-area attribute.

The last thing you need to do to support areas in your app is update the default routing rules for the application in Startup.cs in the Configure method:

```
app.UseMvc(routes =>
{
    // Areas support
    routes.MapRoute(
        name: "areaRoute",
        template: "{area:exists}/{controller=Home}/{action=Index}/{id?}");

    routes.MapRoute(
        name: "default",
        template: "{controller=Home}/{action=Index}/{id?}");
});
```

For example, the sample application for managing various Ninjas, Pirates and so on could utilize Areas to achieve the project organization structure, as shown in **Figure 3**.

The Areas feature provides an improvement over the default convention by providing separate folders for each logical section of the application. Areas are a built-in feature in ASP.NET Core

MVC, requiring minimal setup. If you're not already using them, keep them in mind as an easy way to group related sections of your app together and separate from the rest of the app.

However, the Areas organization is still very folder-heavy. You can see this in the vertical space required to show the relatively small number of files in the Areas folder. If you don't have many controllers per area and you don't have many views per controller, this folder overhead may add friction in much the same way as the default convention.

Fortunately, you can easily create your own convention.

Feature Folders in ASP.NET Core MVC

Outside of the default folder convention or the use of the built-in Areas feature, the most popular way to organize MVC projects is with folders per feature. This is especially true for teams that have adopted

delivering functionality in vertical slices (see bit.ly/2abpJ7t), because most of a vertical slice's UI concerns can exist within one of these feature folders.

When organizing your project by feature (instead of by file type), you'll typically have a root folder (such as Features) within which you'll have a sub-folder per feature. This is very similar to how areas are organized. However, within each feature folder, you'll include all of the required controllers, views and ViewModel types. In most applications, this results in a folder with perhaps five to 15 items in it, all of which are closely related to one another. The entire contents of the feature folder can be kept in view in the Solution Explorer. You can see an example of this organization for the sample project in **Figure 4**.

Notice that even the root-level Controllers and Views folders have been eliminated. The homepage for the app is now in its own feature folder called Home, and shared files like `_Layout.cshtml` are located in a Shared folder within the Features folder, as well. This project organization scales quite well, and lets developers keep their focus on far fewer folders while working on a particular section of an application.

In this example, unlike with Areas, no additional routes are required and no attributes needed for the controllers (note, however, that controller names must be unique between features in this implementation). To support this organization, you need a custom `IViewLocationExpander` and `IControllerModelConvention`. These are both used, along with some custom `ViewLocationFormats`, to configure MVC in your Startup class.

For a given controller, it's useful to know with what feature it's associated. Areas achieve this using attributes; this approach uses a convention. The convention expects the controller to be in a namespace called "Features," and for the next item in the namespace hierarchy after "Features" to be the feature name. This name is added to properties that are available during view location, as shown in **Figure 5**.

You add this convention as part of the `MvcOptions` when adding MVC in Startup:

```
services.AddMvc(o => o.Conventions.Add(new FeatureConvention()));
```

To replace the normal view location logic used by MVC with the feature-based convention, you can clear the list of `ViewLocationFormats` used by MVC and replace it with your own list. This is done as part of the `AddMvc` call, as shown in **Figure 6**.

By default, these format strings include placeholders for actions ("`{0}`"), controllers ("`{1}`"), and areas ("`{2}`"). This approach adds a fourth token ("`{3}`") for features.

The view location formats used should support views with the same name but used by different controllers within a feature. For example, it's quite common to have more than one controller in a feature and for multiple controllers to have an Index method. This is supported by searching for views in a folder matching the controller name. Thus, `NinjasController.Index` and `SwordsController.Index` would locate views in `/Features/Ninjas/Ninjas/Index.cshtml` and `/Features/Ninjas/Swords/Index.cshtml`, respectively (see **Figure 7**).

Note that this is optional—if your features don't have a need to disambiguate views (say, because the feature has only one controller), you can just put the views directly into the feature folder. Also, if you'd rather use file prefixes than folders, you could easily adjust the format string to use "`{3}/{1}`" instead of "`{3}/{1}`," resulting in view filenames like `NinjasIndex.cshtml` and `SwordsIndex.cshtml`.

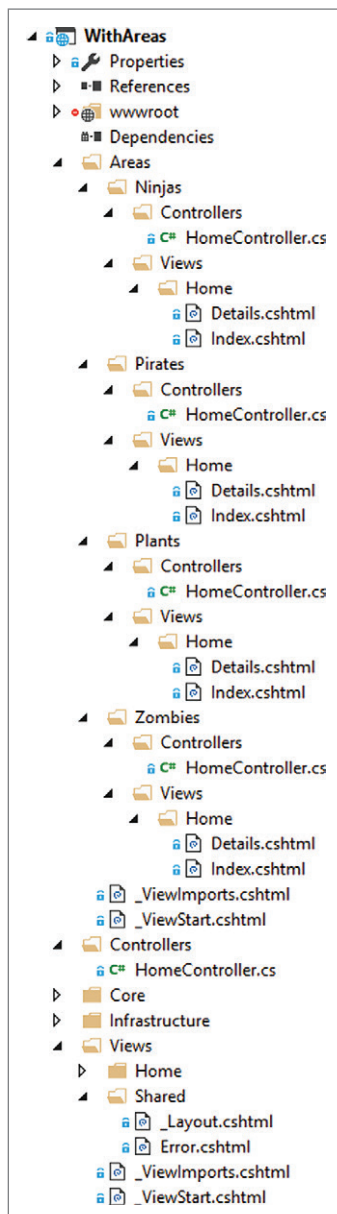


Figure 3 Organizing an ASP.NET Core Project with Areas

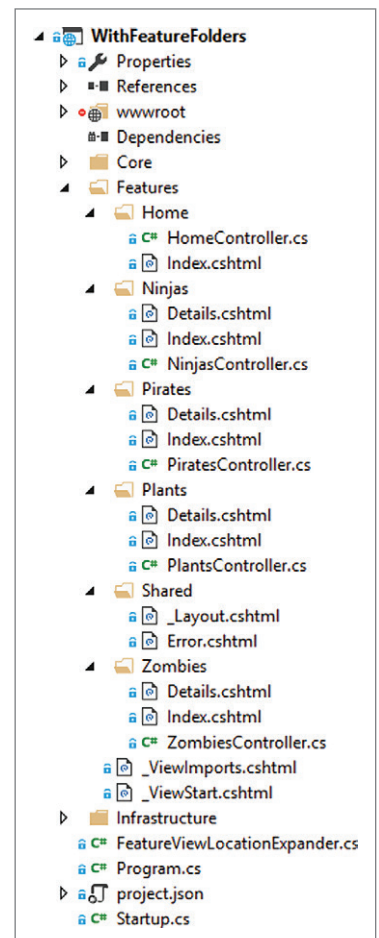


Figure 4 Feature Folder Organization

Shared views are also supported, both in the root of the features folder and in a Shared subfolder.

The `IViewLocationExpander` interface exposes a method, `ExpandViewLocations`, that's used by the framework to identify folders containing views. These folders are searched when an action returns a view. This approach only requires the `ViewLocationExpander` to replace the "{3}" token with the controller's feature name, specified by the `FeatureConvention` described earlier:

```
public IEnumerable<string> ExpandViewLocations(ViewLocationExpanderContext context,
    IEnumerable<string> viewLocations)
{
    // Error checking removed for brevity
    var controllerActionDescriptor =
        context.ActionContext.ActionDescriptor as ControllerActionDescriptor;
    string featureName = controllerActionDescriptor.Properties["feature"] as string;
    foreach (var location in viewLocations)
    {
        yield return location.Replace("{3}", featureName);
    }
}
```

To support publishing correctly, you'll also need to update `project.json`'s `publishOptions` to include the Features folder:

```
"publishOptions": {
  "include": [
    "wwwroot",
    "Views",
    "Areas/**/*.cshtml",
    "Features/**/*.cshtml",
    "appsettings.json",
    "web.config"
  ]
},
```

Figure 5 `FeatureConvention : IControllerModelConvention`

```
{
  public void Apply(ControllerModel controller)
  {
    controller.Properties.Add("feature", GetFeatureName(controller.ControllerType));
  }

  private string GetFeatureName(TypeInfo controllerType)
  {
    string[] tokens = controllerType.FullName.Split('.');
    if (!tokens.Any(t => t == "Features")) return "";
    string featureName = tokens
        .SkipWhile(t => !t.Equals("features",
            StringComparison.CurrentCultureIgnoreCase))
        .Skip(1)
        .Take(1)
        .FirstOrDefault();

    return featureName;
  }
}
```

Figure 6 Replacing the Normal View Location Logic Used by MVC

```
services.AddMvc(o => o.Conventions.Add(new FeatureConvention()))
    .AddRazorOptions(options =>
    {
        // {0} - Action Name
        // {1} - Controller Name
        // {2} - Area Name
        // {3} - Feature Name

        // Replace normal view location entirely
        options.ViewLocationFormats.Clear();
        options.ViewLocationFormats.Add("/{3}/{1}/{0}.cshtml");
        options.ViewLocationFormats.Add("/{3}/{0}.cshtml");
        options.ViewLocationFormats.Add("/{3}/Shared/{0}.cshtml");

        options.ViewLocationExpanders.Add(new FeatureViewLocationExpander());
    })
```

The new convention of using a folder called `Features` is completely under your control, along with how the folders are organized within it. By modifying the set of `ViewLocationFormats` (and possibly the `FeatureViewLocationExpander` type's behavior), you can have full control over where your app's views are located, which is the only thing needed to reorganize your files, because controller types are discovered regardless of the folder in which they're located.

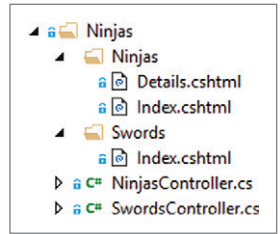


Figure 7 Multiple Controllers Per Feature

Side-By-Side Feature Folders

If you want to try out Feature Folders side-by-side with the default MVC Area and View conventions, you can do so with only small modifications. Instead of clearing the `ViewLocationFormats`, insert the feature formats into the start of the list (note the order is reversed):

```
options.ViewLocationFormats.Insert(0, "/Features/Shared/{0}.cshtml");
options.ViewLocationFormats.Insert(0, "/Features/{3}/{0}.cshtml");
options.ViewLocationFormats.Insert(0, "/Features/{3}/{1}/{0}.cshtml");
```

To support features combined with areas, modify the `AreaViewLocationFormats` collection, as well:

```
optionsAreaViewLocationFormats.Insert(0, "/Areas/{2}/Features/Shared/{0}.cshtml");
optionsAreaViewLocationFormats.Insert(0, "/Areas/{2}/Features/{3}/{0}.cshtml");
optionsAreaViewLocationFormats.Insert(0, "/Areas/{2}/Features/{3}/{1}/{0}.cshtml");
```

What About Models?

Astute readers will notice that I didn't move my model types into the feature folders (or Areas). In this sample, I don't have separate `ViewModel` types, because the models I'm using are incredibly simple. In a real-world app, it's likely your domain or persistence model will have more complexity than your views require, and that it'll be defined in its own, separate project. Your MVC app will likely define `ViewModel` types that contain just the data required for a given view, optimized for display (or consumption from a client's API request). These `ViewModel` types absolutely should be placed in the feature folder where they're used (and it should be rare for these types to be shared between features).

Wrapping Up

The sample includes all three versions of the `NinjaPiratePlant-Zombie` organizer application, with support for adding and viewing each data type. Download it (or view it on [GitHub](#)) and think about how each approach would work in the context of an application you work on today. Experiment with adding an Area or a feature folder to a larger application you work on and decide if you prefer working with feature slices as the top-level organization of your app's folder structure rather than having top-level folders based on file types.

The source code for this sample is available at bit.ly/29Mxsl0. ■

STEVE SMITH is an independent trainer, mentor and consultant, as well as an ASP.NET MVP. He has contributed dozens of articles to the official ASP.NET Core documentation (docs.asp.net), and helps teams quickly get up to speed with ASP.NET Core. Contact him at ardalis.com.

THANKS to the following technical expert for reviewing this article: [Ryan Nowak](#)

Build Asynchronous AJAX-Enabled Web Pages with Reactive Extensions

Peter Vogel

In an earlier article, I discussed how the observer pattern can be used to manage long-running tasks (msdn.com/magazine/mt707526). By the end of that article, I showed how Microsoft Reactive Extensions (Rx) provide a simple mechanism for managing a sequence of events from a long-running process in a Windows application.

Using Rx just for monitoring a sequence of events from a long-running task, however, doesn't take full advantage of the technology. The beauty of Rx is that it can be used for asynchronously integrating any event-based process with any other process. In this article, for example, I'll use Rx to make asynchronous calls to a Web service from a button click in a Web page (a button click is, effectively, a sequence of one event). To use Rx in the client-side Web environment, I'll use Rx for JavaScript (RxJS).

This article discusses:

- Structuring single-page applications
- Loosely coupling application components
- Integrating RxJS with client-side DOM and jQuery
- Integrating client-side code with RESTful services

Technologies discussed:

RxJS, RxJS-DOM, jQuery, AJAX, RESTful Services

Code download available at:

msdn.com/magazine/0916magcode

Rx provides a standard way to abstract a variety of scenarios and manipulate them using a fluent, LINQ-like interface that lets you compose applications from simpler building blocks. Rx lets you both integrate your UI events with back-end processing while, at the same time, keeping them separate—with Rx you can rewrite your UI without having to make corresponding changes to your back end (and vice versa).

Rx provides a standard way to abstract a variety of scenarios and manipulate them using a fluent, LINQ-like interface that lets you compose applications from simpler building blocks.

RxJS also supports a clean separation between your HTML and your code, effectively giving you data binding without requiring special HTML markup. RxJS also builds on existing client-side technologies (jQuery, for example). There's one other benefit to Rx: all Rx implementations tend to look very similar—the RxJS code in

Visit us at
DEVintersection
Las Vegas, Oct 26-27

Enterprise-Proven Distributed Caching

Trusted for over a decade, the easiest
most powerful in-memory data grid to
scale your .NET applications

- ▶ Fast and linearly scalable
- ▶ Enterprise-grade availability
- ▶ Industry-leading ease of use
- ▶ Integrated in-memory computing

Replacing AppFabric Caching?

Try our source-code compatible drop-in.
www.scaleoutsoftware.com/appfabric

Brought to you by the scalability architects

Step up to a battle-tested in-memory data grid that has been hardened by over 425 enterprise customer deployments. ScaleOut's technology makes advanced features such as parallel LINQ query and integrated MapReduce accessible to any .NET developer. Automatic configuration and turnkey global data replication deliver legendary ease-of-use. ScaleOut's world-class support meets the needs of mission-critical applications. Run on premises or in the cloud on Microsoft Azure or Amazon AWS.



ScaleOut Software



Download your free trial today!
www.scaleoutsoftware.com/trial

this article is very much like the Microsoft .NET Framework code I wrote in my previous article. You can leverage the skills you pick up in one Rx environment in any Rx environment.

Getting Started with RxJS

RxJS achieves its goals by abstracting the parts of an application into two groups. Members of the first group are the observables: fundamentally, anything that fires an event. RxJS provides a rich set of operators for creating observables—observables include anything that can be made to appear to fire an event (Rx can convert arrays into event sources, for example). The Rx operators also let you filter and transform the output of events. It might be worthwhile to think of an observable as a processing pipeline for an event source.

Members of the second group are the observers, which accept results from observables and provide processing for three notifications from an observable—a new event (with its associated data), an error or end-of-the-event sequence. In RxJS, an observer can either be an object with functions to handle one or more of the three notifications or just a collection of functions, one for each notification.

To tie these two groups together, an observable subscribes one or more observers to its pipeline.

You can add RxJS to your project through NuGet (in the NuGet library, look for RxJS-All). One warning, though: When I first added RxJS to a project configured with TypeScript, NuGet Manager asked if I also wanted the relevant TypeScript definition files. Clicking Yes added the files and gave me about 400 “duplicate definition” errors. I’ve stopped accepting that option.

In addition, there are a number of Rx support libraries that provide useful plug-ins for RxJS. For example, RxJS-DOM (available through NuGet as RxJS-Bridges-HTML) provides integration both with events in the client-side DOM and with jQuery. That library is essential for creating responsive Web applications with RxJS.

Like most JavaScript plug-ins for Rx, RxJS-DOM hangs its new features off the Rx object that’s central to the RxJS library. RxJS-DOM adds a DOM property to the Rx object that has multiple useful functions, including several that bridge to jQuery-like functions. For example, to make an AJAX call using RxJS-DOM to retrieve JSON objects, you’d use this code:

```
return Rx.DOM.getJSON("...url...")
```

To use both Rx and RxJS-DOM, all you need are these two script tags:

```
<script src="../../Scripts/rx.all.js"></script>
<script src="../../Scripts/rx.dom.js"></script>
```

Using both rx.all.js and rx.dom.js is a heavyweight solution because they include all of the functionality from both libraries. Fortunately, both the Rx and RxJS-DOM divide their functionality over several, lighter libraries so you can incorporate just the libraries holding the functionality you need into your page (the documentation on GitHub will tell you, for any operator, to which library that operator belongs).

Integrating a RESTful Service

A typical scenario in a JavaScript application is that the user clicks on a button to retrieve a result from a Web service by making an asynchronous call to the service. The first step in building an RxJS solution is to convert the click event on the button into an

observable, something the RxJS/DOM/jQuery integration makes easy to do. This code, for example, uses jQuery to retrieve a reference to an element on the form with an id of getCust and then creates an observable from that element’s click event:

```
var getCust = $('#getCust').get(0);
var getCustObservable = Rx.DOM.fromEvent(getCust, "click");
```

The fromEvent function allows you to create an observable from any event on any element. However, RxJS contains several shortcuts for more “popular” events, including the click event. I could, for example, also have used this code to create an observable from a button’s click event:

```
var getCustObservable = Rx.DOM.click(getCust);
```

I can create a richer processing pipeline for this event that simplifies processing in my application. For example, I might want to avoid handling a scenario where the user clicks the button twice in rapid succession (too quickly, for example, for me to disable the button to prevent the user from doing that). Rather than write a bunch of timeout code, I can handle that by adding the debounce function to the pipeline, specifying that I only want to see clicks that are at least two seconds apart:

```
var getCustObservable = Rx.DOM.click(getCust).debounce(2000);
```

I can also perform some processing on the event using, for example, flatMapLatest, which lets me incorporate a transformation function into the pipeline (something like LINQ SelectMany). The base function—flatMap—performs a transformation on each event in a sequence. The flatMapLatest function goes one step further and handles a typical problem with asynchronous processing: flatMapLatest cancels earlier asynchronous processing if the transformation is invoked a second time and an asynchronous request is still pending.

With RxJS, there’s no need to provide callbacks for this request; the results of the call are automatically passed through the observable to any observers.

With flatMapLatest, if I do have a user who clicks the button twice (and more than two seconds apart) then if flatMapLatest is still transforming the previous event, flatMapLatest will cancel the previous event. This eliminates the need for me to write code to handle canceling asynchronous processing.

The first step in using flatMapLatest is to create my transformation function. To do that, I just wrap the getJSON call I showed earlier inside a function. Because my function is tied to a button click, I don’t need any data from the page (in fact, this function barely qualifies as a “transformation” because it ignores the event’s inputs).

Here’s a function that makes a request to a Web API service, using some jQuery along the way to retrieve the customer Id from an element on the page:



SPECIAL
PULL-OUT
SECTION

Orlando

2016

ROYAL PACIFIC RESORT AT UNIVERSAL
DECEMBER 5-9



The Ultimate Education Destination

Live! 360SM is a unique conference where the IT and Developer community converge to debate leading edge technologies and educate themselves on current ones. These six co-located events incorporate knowledge transfer and networking, along with finely tuned education and training, as you create your own custom conference, mixing and matching sessions and workshops to best suit your needs. **Choose the ultimate education destination: Live! 360.**

LIVE360EVENTS.COM

EVENT PARTNERS



PLATINUM SPONSORS



GOLD SPONSORS



SILVER SPONSOR





SUPPORTED BY



PRODUCED BY



Live! 360 Agenda-at-a-Glance

		Visual Studio 							SQL Server 					TECHMENTOR							
		VISUAL STUDIO LIVE! TRACKS							SQL SERVER LIVE! TRACKS					TECHMENTOR TRACKS							
		ALM / DevOps	Cloud Computing	Mobile Client	Software Practices	Visual Studio / .NET Framework	Web Client	Web Server	Windows Client	BI, Big Data, Data Analytics, and Data Visualization	SQL Server Administration & Maintenance	SQL Server in the Cloud	SQL Server for Developers	SQL Server Performance Tuning and Optimization	Client	DevOps	IT Soft Skills	The Real Cloud	Security	Server / Datacenter	
START TIME	END TIME	Visual Studio Live! Pre-Conference: Sunday, December 4, 2016							SQL Server Live! Pre-Conference: Sunday, December 4, 2016					TechMentor Pre-Conference: Sunday, December 4, 2016							
5:00 PM	8:00 PM	Pre-Conference Registration • Royal Pacific Resort Conference Center							Pre-Conference Registration • Royal Pacific Resort Conference Center					Pre-Conference Registration • Royal Pacific Resort Conference Center							
6:00 PM	9:00 PM	Dine-A-Round Dinner @ Universal CityWalk (6:00pm—Meet at Conference Registration Desk to walk over with the group)							Dine-A-Round Dinner @ Universal CityWalk (6:00pm—Meet at Conference Registration Desk to walk over with the group)					Dine-A-Round Dinner @ Universal CityWalk (6:00pm—Meet at Conference Registration Desk to walk over with the group)							
START TIME	END TIME	Visual Studio Live! Pre-Conference Workshops: Monday, December 5, 2016							SQL Server Live! Pre-Conference Workshops: Monday, December 5, 2016					TechMentor Pre-Conference Workshops: Monday, December 5, 2016							
8:00 AM	5:00 PM	VSM01 - Workshop: Distributed Cross-Platform Application Architecture - Rockford Lhotka & Jason Bock		VSM02 - Workshop: Service Oriented Technologies—Designing, Developing, & Implementing WCF and the Web API - Miguel Castro			VSM03 - Workshop: DevOps in a Day - Brian Randell		SQM01 - Workshop: Performance Tune SQL Server: Query Optimizer, Indexes, the Plan Cache, and Execution Plans - Bradley Ball		SQM02 - Workshop: SQL Server 2016 for Developers - Leonard Lobel			TMM01 - Workshop: Sixty-seven VMware vSphere Tricks That'll Pay for This Conference! - Greg Shields			TMM02 - Workshop: Demystifying the Blue Screen of Death - Bring Your Own Laptop Hands-On Lab (BYOL - HOL) - Bruce Mackenzie-Low				
5:00 PM	6:00 PM	EXPO Preview																			
6:00 PM	7:00 PM	LIVE! 360 KEYNOTE: To Be Announced																			
START TIME	END TIME	Visual Studio Live! Day 1: Tuesday, December 6, 2016							SQL Server Live! Day 1: Tuesday, December 6, 2016					TechMentor Day 1: Tuesday, December 6, 2016							
8:00 AM	9:00 AM	VISUAL STUDIO LIVE! KEYNOTE: To Be Announced							SQL SERVER LIVE! KEYNOTE: To Be Announced					TECHMENTOR KEYNOTE: Sweet Sixteen, or Just Server 2012R3? A Glance at the Awesome, the Irritating, the Improved and the Expensive in Server 2016 - Mark Minasi, IT Consultant, Author, Speaker, MR&D							
9:00 AM	9:30 AM	Networking Break • Visit the EXPO																			
9:30 AM	10:45 AM	VST01 - Building Applications with ASP.NET Core - Scott Allen		VST02 - Busy .NET Developer's Guide to Swift - Ted Neward		VST03 - What's New in Azure v2 - Eric D. Boyd		VST04 - Real World Scrum with Team Foundation Server 2015 & Visual Studio Team Services - Benjamin Day		SQT01 - Performance Tuning and Monitoring for Virtualized Database Servers - Thomas LaRock		SQT02 - What's New in SQL Server 2016 - Leonard Lobel		SQT03 - Power BI: Analytics for Desktop, Mobile and Cloud - Andrew Brust		TMT01 - Secure Access Everywhere! Implementing DirectAccess in Windows Server 2016 - Richard Hicks		TMT02 - Linux on Azure for the Microsoft Specialist - Timothy Warner		TMT03 - The Absolute Beginner's Guide to Advanced Certificate Services - Greg Shields	
11:00 AM	12:15 PM	VST05 - Richer MVC Sites with Knockout JS - Miguel Castro		VST06 - Busy .NET Developer's Guide to Native iOS - Ted Neward		VST07 - Overview of Power Apps - Nick Pinheiro		VST08 - Get Good at DevOps: Feature Flag Deployments with ASP.NET, WebAPI, & JavaScript - Benjamin Day		SQT04 - U-SQL Query Execution and Performance Tuning - Michael Rys		SQT05 - Implementing Data Protection and Security in SQL Server 2016 - Steve Jones		SQT06 - Seeking the Perfect Hybrid: On-Prem Data with Reports and Dashboards in Power BI - William E. Pearson III		TMT04 - DirectAccess Troubleshooting Deep Dive - Richard Hicks		TMT05 - Container Technology and its Impact on Datacenter and Cloud Management - Neil Peterson		TMT06 - Master Camtasia and Build Your Own Training in 75 Minutes or Less - Greg Shields	
12:15 PM	2:00 PM	Lunch • Visit the EXPO																			
2:00 PM	3:15 PM	VST09 - WCF & Web API: Can We All Just Get Along?!! - Miguel Castro		VST10 - Creating Great Looking Android Applications Using Material Design - Kevin Ford		VST11 - Introduction to Next Generation of Azure PaaS—Service Fabric and Containers - Vishwas Lele		VST12 - To Be Announced		SQT07 - New Performance Tuning and Security Features in SQL Server 2016 - Thomas LeBlanc		SQT08 - Introduction to Elastic Azure SQL Database - Bradley Ball		SQT09 - To Be Announced		TMT07 - Windows as a Service Explained: Really, I've Got to Upgrade Every Year? - Mark Minasi		TMT08 - To Be Announced		TMT09 - Getting Started with Nano Server - Jeffery Hicks	
3:15 PM	4:15 PM	Networking Break • Visit the EXPO																			
4:15 PM	5:30 PM	VST13 - Busy Developer's Guide to Chrome Development - Ted Neward		VST14 - Using Visual Studio Tools for Apache Cordova to Create MultiPlatform Applications - Kevin Ford		VST15 - Cloud Oriented Programming - Vishwas Lele		VST16 - Bringing DevOps to the Database - Steve Jones		SQT10 - Performance Tuning and Troubleshooting Azure SQL Database - Bradley Ball		SQT11 - No Schema, No Problem! Introduction to Azure DocumentDB - Leonard Lobel		SQT12 - Getting Started with Data Analysis eXpressions (DAX) in Analysis Services Tabular 2016 - William E. Pearson III		TMT10 - Troubleshooting Client Communications with Wireshark - Timothy Warner		TMT11 - Azure Point-to-Site VPN Suck! Fix It with Win2012R2 VPN in the Cloud - Richard Hicks		TMT12 - Implementing Hyper-V Failover Clusters in Windows Server 2016 - Bruce Mackenzie-Low	
5:30 PM	7:30 PM	Exhibitor Reception																			
START TIME	END TIME	Visual Studio Live! Day 2: Wednesday, December 7, 2016							SQL Server Live! Day 2: Wednesday, December 7, 2016					TechMentor Day 2: Wednesday, December 7, 2016							
8:00 AM	9:15 AM	VSW01 - Moving from Angular 1 to Angular 2 - Ben Dewey		VSW02 - The Future of Mobile Application Search - James Montemagno		VSW03 - Managing Enterprise and Consumer Identity with Azure Active Directory - Nick Pinheiro		VSW04 - Improving Performance in .NET Applications - Jason Bock		SQW01 - Inside the SQL Server Query Optimizer: SQL 2014 & 2016 - Bradley Ball		SQW02 - SQL Server 2016 Encryption - Thomas LaRock		SQW03 - Big Data with Hadoop, Spark and Azure HDInsight - Andrew Brust		TMW01 - Creating Advanced Functions in PowerShell - Michael Wiley		TMW02 - Implementing Azure AD for Hybrid Identity - Timothy Warner		TMW03 - Managing Windows 10 Using the MDM Protocol and ConfigMgr - Steven Rachui	
9:30 AM	10:45 AM	VSW05 - Getting Started with Aurelia - Brian Noyes		VSW06 - Building Connected and Disconnected Mobile Applications - James Montemagno		VSW07 - Practical Internet of Things for the Microsoft Developer - Eric D. Boyd		VSW08 - I'll Get Back to You: Understanding Task, Await, and Asynchronous Methods - Jeremy Clark		SQW04 - Does Your Performance Tuning Need a 12-Step program? - Janis Griffin		SQW05 - Master Data Management with Data Quality (DQS) and Master Data Services (MDS) in SQL Server 2016 - Thomas LeBlanc		SQW06 - Getting Started with PolyBase in SQL Server 2016 - Edwin Sarmiento		TMW04 - Creating Class-Based PowerShell Tools - Jeffery Hicks		TMW05 - Fully Integrated Azure Resource Manager Deployments - Neil Peterson		TMW06 - Building Applications in ConfigMgr—Tips and Tricks - Steven Rachui	
10:45 AM	11:15 AM	Networking Break • Visit the EXPO																			
11:15 AM	12:15 PM	LIVE! 360 KEYNOTE: To Be Announced																			
12:15 PM	1:45 PM	Birds-of-a-Feather Lunch • Visit the EXPO																			
1:45 PM	3:00 PM	VSW09 - Living in a Command Line Web Development World (NPM, Bower, Gulp, and More) - Ben Dewey		VSW10 - Understanding the Windows Desktop App Development Landscape - Brian Noyes		VSW11 - To Be Announced		VSW12 - Learn to Love Lambdas (and LINQ, Too) - Jeremy Clark		SQW07 - Configuring SQL Server for Performance—Like a Microsoft Certified Master - Thomas LaRock		SQW08 - To Be Announced		SQW09 - Big Data's Missing V: Visualization. How Do You BigViz Your Big Data? - Jennifer Stirrup		TMW07 - Creating WPF-Based Graphical PowerShell Tools - Jeffery Hicks		TMW08 - To Be Announced		TMW09 - SysInternals Tools: Process Explorer and Process Monitor - Sami Laiho	
3:00 PM	4:00 PM	Networking Break • Visit the EXPO • Expo Raffle @ 3:30 p.m.																			
4:00 PM	5:15 PM	VSW13 - Securing Client JavaScript Apps - Brian Noyes		VSW14 - Let's Write a Windows 10 App: A Basic Introduction to Universal Apps - Billy Hollis		VSW15 - ARM Yourself for Azure Success - Esteban Garcia		VSW16 - Continuous Delivery on Azure: A/B Testing, Canary Releases, and Dark Launching - Marcel de Vries		SQW10 - Performance in 60 Seconds—SQL Tricks Everybody MUST Know - Pinal Dave		SQW11 - Welcome to the 2016 Query Store! - Janis Griffin		SQW12 - Improve Enterprise Reporting with SQL Server Analysis Services - Thomas LeBlanc		TMW10 - Harvesting the Web: Using PowerShell to Scrape Screens, Exploit Web Services, and Save Time - Mark Minasi		TMW11 - In-Depth Introduction to Docker - Neil Peterson		TMW12 - War Driving: How it Happens, How to Protect Yourself - Dale Meredith	
8:00 PM	10:00 PM	Live! 360 Dessert Luau																			
START TIME	END TIME	Visual Studio Live! Day 3: Thursday, December 8, 2016							SQL Server Live! Day 3: Thursday, December 8, 2016					TechMentor Day 3: Thursday, December 8, 2016							
8:00 AM	9:15 AM	VSH01 - Build Real-Time Websites and Apps with SignalR - Rachel Appel		VSH02 - Cognitive Services: Building Smart Applications with Computer Vision - Nick Landry		VSH03 - C# Best Practices - Scott Allen		VSH04 - Application Insights: Measure Your Way to Success - Esteban Garcia		SQH01 - Stretching SQL Server Failover Clustered Instances & Availability Groups to a Disaster Recovery Site - Edwin Sarmiento		SQH02 - Production SQL Server 2016—Lessons from the Field - Joseph D'Antoni		SQH03 - Agile Analytics with AzureML and R - Jennifer Stirrup		TMH01 - PowerShell and Workflow—Magic Together! - Michael Wiley		TMH02 - Pen-Testing Like an IT Superhero - Dale Meredith		TMH03 - Windows Clusters for Beginners: From Highly Fearful to Highly Reliable in 75 Minutes! - Mark Minasi	
9:30 AM	10:45 AM	VSH05 - HTTP/2: What You Need to Know - Robert Boedigheimer		VSH06 - Building Business Apps on the Universal Windows Platform - Billy Hollis		VSH07 - Debugging Your Way Through .NET with Visual Studio 2015 - Ida Flatow		VSH08 - The Ultimate Intro to Docker for Developers - Adam Tuliper		SQH04 - Indexes—The Good, Bad and Ugly - Pinal Dave		SQH05 - Would You Just Load Already?! Maximizing Your SSIS Data Load - Chris Bell		SQH06 - A Blueprint for Business Intelligence with SQL Server 2016 - Jennifer Stirrup		TMH04 - You're Writing Your PowerShell Functions Wrong. Stop It. - Don Jones		TMH05 - Mobile Devices and Security: The Bane of the IT Superhero - Dale Meredith		TMH06 - Office 365 Risk Mitigation - J. Peter Bruzzese	
11:00 AM	12:15 PM	VSH09 - TypeScript and ES2015 JumpStart - John Papa		VSH10 - HoloLens - Billy Hollis & Brian Randell		VSH11 - Exploring Microservices in a Microsoft Landscape - Marcel de Vries		VSH12 - Automated UI Testing for iOS and Android Mobile Apps - James Montemagno		SQH07 - Securing Your Database with SQL Server 2016 Features - Edwin Sarmiento		SQH08 - Statistics and Cardinality—How They Work Together to Find Data Efficiently - Chris Bell		SQH09 - New and Shiny: The All-New SQL Server Reporting Services 2016 - Tim Mitchell		TMH07 - PowerShell Desired State Configuration (DSC) for the IT Ops Guy - Jason Helmick		TMH08 - Understanding Windows 10/2016's Super Security: VSM, Credential Guard, Trustlets and More - Mark Minasi		TMH09 - Evolving as an IT Pro - J. Peter Bruzzese	
12:15 PM	1:30 PM	Lunch on the Lanai																			
1:30 PM	2:45 PM	VSH13 - All Your Tests Are Belong To Us - Rachel Appel		VSH14 - Developing Awesome 3D Apps with Unity and C# - Adam Tuliper		VSH15 - Unit Testing Makes Me Faster: Convincing Your Boss, Your Co-Workers, and Yourself - Jeremy Clark		VSH16 - Writing Maintainable, X-Browser Automated Tests - Marcel de Vries		SQH10 - Secrets of SQL Server—Database Worst Practices - Pinal Dave		SQH11 - Powerful T-SQL Improvements that Reduce Query Complexity - Hugo Kornelis		SQH12 - Testing SSIS Packages - Tim Mitchell		TMH10 - PowerShell Unplugged: - Don Jones		TMH11 - Facing Increasing Malware Threats and a Growing Trend of BYOD with a New Approach of PC Security - Yung Chau		TMH12 - The Labyrinth of Exchange Migration Options - J. Peter Bruzzese	
3:00 PM	4:15 PM	VSH17 - SASS and CSS for Developers - Robert Boedigheimer		VSH18 - From Oculus to HoloLens: Building Virtual & Mixed Reality Apps & Games - Nick Landry		VSH19 - User Experience Case Studies—Good and Bad - Billy Hollis		VSH20 - Debugging the Web with Fiddler - Ida Flatow		SQH13 - Hacking Expose—Using SSL to Secure SQL Server Connections - Chris Bell		SQH14 - T-SQL User-Defined Functions, or: Bad Performance Made Easy - Hugo Kornelis		SQH15 - Essential Architecture for BI in a Virtual Environment - Joseph D'Antoni		TMH13 - Building Highly Available and Secure DSC Pull Servers - Jason Helmick		TMH14 - Penetration Tests in Real Life - Sami Laiho		TMH15 - What's IoT Got to Do with IT Pros? A Lot! - Yung Chau	
4:30 PM	5:30 PM	Live! 360 Conference Wrap-Up - Andrew Brust (Moderator), Andrew Connell, Don Jones, Rockford Lhotka, Matthew McDermott, Brian Randell, & John K. Waters																			
START TIME	END TIME	Visual Studio Live! Post-Conference Workshops: Friday, December 9, 2016							SQL Server Live! Post-Conference Workshops: Friday, December 9, 2016					TechMentor Post-Conference Workshops: Friday, December 9, 2016							
8:00 AM	5:00 PM	VSF01 - Workshop: Angular 2 Bootcamp - John Papa				VSF02 - Workshop: Building Modern Web Apps with Azure - Eric D. Boyd & Brian Randell			SQF01 - Workshop: Design and Implement SQL Server HA/DR Hybrid Solutions with Microsoft Azure - Edwin Sarmiento			SQF02 - Workshop: Big Data, Analytics and NoSQL: Everything You Wanted to Learn But Were Afraid to Ask - Andrew Brust		TMF01 - Workshop: Boost Your IT Career, 2017 Edition: The Don and Jason Show - Don Jones and Jason Helmick			TMF02 - Workshop: BlackBelt—Windows Security Internals - Sami Laiho				

Office & SharePoint <small>LIVE!</small>				ModernApps <small>LIVE!</small>	APPDEV TRENDS <small>NEW!</small>					
OFFICE & SHAREPOINT LIVE! TRACKS				MODERN APPS LIVE! TRACK	APP DEV TRENDS TRACKS					
SharePoint On-Premises Infrastructure Management and Administration	SharePoint, Office 365 and the Cloud	High-Value SharePoint Workloads: Social, Search, BI, Workflow, and Business Process Automation	Developing for Office, Office 365 and SharePoint	Presented in Partnership with: Magenic	Agile	Containerization	Continuous Integration	Java	Mobile	Cloud
Office & SharePoint Live! Pre-Conf. Sun., Dec. 4, 2016				MAL! Pre-Conf.: Sun., Dec. 4	ADT Pre-Conf.: Sun., Dec. 4					
Pre-Conference Registration • Royal Pacific Resort Conference Center										
Dine-A-Round Dinner @ Universal CityWalk (6:00pm—Meet at Conference Registration Desk to walk over with the group)										
Office & SharePoint Live! Mon., Dec. 5, 2016				MAL! Monday, Dec. 5	ADT: Pre-Conference Workshops: Monday, Dec. 5					
8:00 AM – 12:00 PM: OSM01 - Workshop: Installing and Configuring SharePoint Server 2016 - Vlad Catrinescu	8:00 AM – 12:00 PM: OSM02 - Workshop: A Beginner's Guide to Client Side Development in SharePoint - Mark Rackley			MAM01 - Workshop: Building Modern Mobile Apps - Brent Edwards & Kevin Ford	ADM01 - Workshop: Building Teams - Steve Green		ADM02 - Workshop: One Codebase to Rule Them All: Xamarin - Fabian Williams			
1:00 PM – 5:00 PM: OSM03 - Workshop: Upgrade to SharePoint 2016 - Matthew McDermott	1:00 PM – 5:00 PM: OSM04 - Workshop: Building Clients Side Applications in Office 365 with the New SharePoint Framework - Andrew Connell									
Office & SharePoint Live! Day 1: Tues., Dec. 6				MAL! Day 1: Tues., Dec. 6	ADT Day 1: Tues., Dec. 6					
OFFICE & SHAREPOINT LIVE! KEYNOTE: The Modern Workplace and the SharePoint Revolution - Chris Bortlik, Collaboration Solution Architect, Microsoft				MODERN APPS LIVE! KEYNOTE: To Be Announced	APP DEV TRENDS KEYNOTE: To Be Announced					
OST01 - What's New in SharePoint 2016 for IT Pros - Vlad Catrinescu	OST02 - To Be Announced			MAT01 - Modern App Development: Transform How You Build Web and Mobile Software - Rockford Lhotka	ADM01 - Hacking Technical Debt - Steve Green		ADT02 - Java 8 Lambdas and the Streaming API - Michael Remijan			
OST03 - Optimizing SQL Server for SharePoint - Brian Alderman	OST04 - How It Works: Office 365 and the Microsoft Graph - Bill Ayers			MAT02 - Architecture: The Key to Modern App Success - Brent Edwards	ADT03 - Are You A SOLID Coder? - Steve Green		ADT04 - PrimeFaces 5: Modern UI Widgets for Java EE - Kita Mann			
OST05 - IT Pros Guide to Managing SharePoint Search - Matthew McDermott	OST06 - To Be Announced			MAT03 - Manage Distributed Teams with Visual Studio Team Services and Git - Brian Randell	ADT05 - Agile Architecture - Steve Green		ADT06 - Full Stack Java with JSweet, Angular 2, PrimeNG, and JAX-RS - Kita Mann			
OST07 - Integrating Office Online Server with SharePoint - Brian Alderman	OST08 - Introduction to the Office Dev PnP Core Library - Rob Windsor			MAT04 - Focus on the User Experience #FTW - Anthony Handley	ADT07 - Crafting Innovation - Steve Green		ADT08 - Who's Taking Out the Garbage? How Garbage Collection Works in the VM - Kita Mann			
Office & SharePoint Live! Day 2: Wed., Dec. 7				MAL! Day 2: Wed., Dec. 7	ADT Day 2: Wed., Dec. 7					
OSW01 - PowerShell for Office 365 - Vlad Catrinescu	OSW02 - Become a Developer Hero by Building Office Add-ins - Bill Ayers			MAW01 - DevOps, Continuous Integration, the Cloud, and Docker - Dan Nordquist	ADW01 - Stop Killing Requirements! - Melissa Green		ADW02 - Migrating Customers to Microsoft Azure: Lessons Learned From the Field - Ida Flatow			
OSW03 - Managing Data Recovery in SharePoint - Brian Alderman	OSW04 - Utilizing jQuery in SharePoint—Get More Done Faster - Mark Rackley			MAW02 - Mobile Panel - Kevin Ford, Rockford Lhotka, James Montemagno, & Ryan J. Salva	ADW03 - Meeting-Free Software Development in Distributed Teams - Yegor Bugayenko		ADW04 - The Essentials of Building Cloud-Based Web Apps with Azure - Ida Flatow			
OSW05 - Scripting SharePoint 2016 Tasks with PowerShell - Ben Stegink	OSW06 - Customizing Your SharePoint Forms Without Third Party Applications - Mark Rackley			MAW03 - C# Everywhere: How CSLA .NET Enables Amazing Cross-Platform Code Reuse - Rockford Lhotka	ADW05 - Introduction to Microsoft Office Graph - Fabian Williams		ADW06 - Building IoT and Big Data Solutions on Azure - Ida Flatow			
OSW07 - Learn Best Practices for Managing and Administering SharePoint Online and OneDrive for Business - Chris Bortlik	OSW08 - Using the Office UI Fabric - Paul Schaefflein			MAW04 - Coding for Quality and Maintainability - Jason Back	ADW07 - As You Think About Azure Databases, Think About DocumentDb - Fabian Williams		ADW08 - Where Does JavaScript Belong in the App Store? - Jordan Matthiesen			
Office & SharePoint Live! Day 3: Thurs., Dec. 8				MAL! Day 3: Thurs., Dec. 8	ADT Day 3: Thurs., Dec. 8					
OSH01 - Implementing and Managing Office 365 - Ben Stegink	OSH02 - Enterprise JavaScript Development Patterns - Rob Windsor			MAH01 - Modern Mobile Development: Build a Single App For iOS & Android with Xamarin Forms - Kevin Ford	ADH01 - From VMs to Containers: Introducing Docker Containers for Linux and Windows Server - Ida Flatow		ADH02 - Continuous Testing in a DevOps World - Wayne Ariola			
OSH03 - Five Business Challenges of Hybrid Search in SharePoint 2016 and Office 365 - Agnes Molnar	OSH04 - Made for Mobile—Let the Microsoft Graph Power Your Mobile Apps! - Bill Ayers			MAH02 - Universal Windows Development: UWP for PC, Tablet & Phone - Brent Edwards	ADH03 - CQRS 2.0 - Commands, Actors, and Events...Oh My! - David Hoerster		ADH04 - Microservices as Chat Bots are the Future - Yegor Bugayenko			
OSH05 - To the Cloud! Using IaaS as a Hosting Provider for SharePoint - Scott Hoag & Dan Usher	OSH06 - Introduction to the SharePoint Client Object Model and REST API - Rob Windsor			MAH03 - Modern Web Development: ASP.NET MVC and Web API - Allen Conway	ADH05 - The Curious Case for the Immutable Object - David Hoerster		ADH06 - Continuous Integration May Have Negative Effects - Yegor Bugayenko			
OSH07 - Findability in YOUR Organization - Agnes Molnar	OSH08 - Leveraging Angular2 to Build Office Add-ins - Andrew Connell			MAH04 - Modern Web Development: Building a Smart Web Client with TypeScript and Angular2 - Allen Conway	ADH07 - To Be Announced		ADH08 - Mobile DevOps Demystified with Xamarin, VSTS and HockeyApp - Roy Cornelissen			
OSH09 - Setting Up Directory Synchronization for Office 365 - Scott Hoag & Dan Usher	OSH10 - Webhooks in Office 365 - Paul Schaefflein			MAH05 - Using All That Data: Power BI to the Rescue - Scott Diehl	ADH09 - Get Started with Microsoft PowerApps - Fabian Williams		ADH10 - Overcoming the Challenges of Mobile Development in the Enterprise - Roy Cornelissen			
O&SPL! Post-Conf. Workshop: Fri., Dec. 9				MAL! Fri., Dec. 9	ADT Post-Conf. Workshop: Fri., Dec. 9					
OSF01 - Workshop: 10 Steps to be Successful with Enterprise Search - Agnes Molnar				MAF01 - Workshop: Modern App Deep Dive: Xamarin, Responsive Web, UWP, CSLA .NET - Jason Back, Allen Conway, Brent Edwards & Kevin Ford	ADF01 - Workshop: To Be Announced					

Take the Tour

Six events, 25+ tracks, and literally hundreds of sessions to choose from – mix and match sessions to create your own, custom event line-up – it's like no other conference available today.

Visual Studio LIVE!

EXPERT SOLUTIONS FOR .NET DEVELOPERS

Visual Studio Live!: VSLive!™ is a victory for code, featuring unbiased and practical development training on the Microsoft Platform.

SQL Server LIVE!

TRAINING FOR DBAs AND IT PROS

SQL Server Live!: This conference will leave you with the skills needed to Lead the Data Race, whether you are a DBA, developer, IT Pro, or Analyst.

TECHMENTOR

IN-DEPTH TRAINING FOR IT PROS

TechMentor™: This is IT training that finishes first, with zero marketing speak on topics you need training on now, and solid coverage on what's around the corner.

Office & SharePoint LIVE!

ON-PREMISE, CLOUD & CROSS-PLATFORM TRAINING

Office & SharePoint Live!: Provides leading-edge knowledge and training for SharePoint both on-premises and in Office 365 to maximize the business value.

ModernApps LIVE!

MOBILE, CROSS-DEVICE & CLOUD DEVELOPMENT

Modern Apps Live!: Presented in partnership with Magenic, this unique conference leads the way to learning how to architect, design and build a complete Modern App.

NEW! APPDEV TRENDS

ENTERPRISE FOCUSED. CODE DRIVEN.

App Dev Trends: This new technology conference focuses on the makers & maintainers of the software that power organizations in nearly every industry in the world– in other words, enterprise software professionals!

LIVE! 360

TECH EVENTS WITH PERSPECTIVE

Featured Speakers

Live! 360 brings you over 90 speakers; some of the best and brightest experts in the industry.

[VIEW THE FULL LIST OF SPEAKERS AT LIVE360EVENTS.COM](http://LIVE360EVENTS.COM)

Visual Studio LIVE!



Rachel Appel



Jason Bock



Billy Hollis



Miguel Castro



Marcel de Vries



Adam Tuliper



Nick Landry



Brian Noyes



John Papa



Brian Randall

SQL Server LIVE!



Andrew Brust



Pinal Dave



Edwin Sarmiento



Leonard Lobel



Janis Griffin

TECHMENTOR



Jeffery Hicks



Don Jones



Sami Laiho



Mark Minasi



Greg Shields

Office & SharePoint LIVE!



Bill Ayers



Mark Rackley



Andrew Connell



Matthew McDermott



Agnes Molnar

ModernApps LIVE!



Allen Conway



Brent Edwards



Kevin Ford



Scott Diehl



Rockford Lhotka

APPDEV TRENDS NEW!



Steven Green



Melissa Green



Kito D Mann



Fabian Williams



Yegor Bugayenko

6 Great Conferences
1 Great Price

Connect with Live! 360



twitter.com/live360
@live360



facebook.com
Search "Live 360"



linkedin.com
Join the "Live! 360" group!

REGISTER BY OCTOBER 5
AND SAVE \$400!



Use promo code
L360TipIn

Scan the QR code to
register or for more
event details.

LIVE360EVENTS.COM

REGISTER BY OCTOBER 5
AND SAVE \$400!

Use promo code L360TipIn



CONNECT WITH LIVE! 360

twitter.com/@live360events

facebook.com - Search "Live 360"

linkedin.com - Join the "Live 360" group!


```
function getCustomer()
{
    return Rx.DOM.getJSON("api/customerorders/customerbyid/" + $("#custId").val());
}
```

With RxJS, there's no need to provide callbacks for this request; the results of the call are automatically passed through the observable to any observers.

To integrate this transformation into my processing chain, I just call `flatMapLatest` from my observer, passing a reference to the function:

```
var getCustObsvble = Rx.DOM.click(getCust).debounce(2000).flatMapLatest(getCustomer);
```

I must now subscribe processing functions to the observable. I can assign up to three functions in the subscription: one function to handle the notification when a new event is received, one to report on any error notifications and one to handle a notification sent at the end of the sequence of events. Because my click event processing pipeline is designed specifically to produce a sequence of one event, I don't need to provide an "end of sequence" function.

The resulting code to assign the two necessary functions might look like this:

```
var getCustObsvble.subscribe(
    c => $("#custName").val(c.FirstName),
    err => $("#Message").text("Unable to retrieve Customer: " + err.description)
);
```

If I thought I might find this set of functions useful in other places (or just wanted to simplify my subscribe code), I could create an observer object. An observer object has the same functions that I used before, just assigned to properties named `onNext`, `onError` and `onComplete`. Because I don't need to handle `onComplete`, my observer object would look like this:

```
var custObserver = {
    onNext: c => $("#custName").val(c.FirstName),
    onError: err => $("#Message").text("Unable to retrieve Customer: " +
    err.description)
};
```

With a separate observer, the code my observable uses to subscribe the observer becomes simpler:

```
var getCustObsvble.subscribe(custObserver);
```

Putting all of this together, I just need a ready function for the page that retrieves the button, attaches the pipeline that turns the button into a very sophisticated observable and then subscribes an observer to that pipeline. Because RxJS implements a fluent interface, I could do it in just two lines of code: one line of jQuery code to retrieve the button element and another line of RxJS code to build the pipeline and subscribe my observer. However, breaking the RxJS pipeline construction and subscribe code over two lines will make the ready function easier for the next programmer to read.

The final version of my ready function would look like this:

```
$(function () {
    var getCust = $('#getButton').get(0);
    var getCustObsvble =
        Rx.DOM.click(getCust).debounce(2000).flatMapLatest(getCustomer);
    getCustObsvble.subscribe(custObserver);
});
```

As a bonus, this process that integrates the click event, the call to the Web service and the data display executes asynchronously. RxJS abstracts all the ugly details of making that work away from me.

Abstracting Event Sequences

By abstracting the process (and providing a rich set of operators) Rx makes many things look alike. This lets you make significant

changes to your process without having to make significant structural changes to your code.

For example, because RxJS-DOM treats an observable with one event (a button click) much like an observable that generates a continuous sequence of events (like mousemove), I can make a significant change to my UI without having to do much to my code. For example, I might decide that, rather than have the user trigger the Web service request with a button click, I'll retrieve the customer data as soon as the user types in the customer's Id.

By abstracting the process (and providing a rich set of operators) Rx makes many things look alike.

The first thing I need to change is the element whose events I'm going to observe. In this case, that means switching from the button on the page to the textbox on the page that holds the customer Id (I'll also change the name of the variable holding the element):

```
var getCustId = $('#custId').get(0);
```

I could turn any number of events for this textbox into an observable. If, for example, I used the blur event on the textbox, I would make my call to the Web service when the user exits the textbox. However, I want to be more responsive than that. I instead choose to retrieve the customer object as soon as the user types "enough" characters into the textbox. That means switching to the `keyup` event, which generates a sequence of events: one for every keystroke.

That change to my pipeline looks like this:

```
var getCustObsvble = Rx.DOM.keyup(getCustId)
```

As the user types in characters, I could end up calling my transformation function multiple times. In fact, if the user types faster than I can retrieve Customer objects, I'm going to end up with multiple stacked requests. I could count on `flatMapLatest` to clean up those requests, but there's a better solution: In my Customer Orders Management system, a customer Id is always four characters long. As a result, there's no point in calling my transformation function except when there are exactly four characters in the textbox (and, for the record, because my pipeline is now taking customer Id and returning a complete Customer object, my transformation function is now actually doing a "transformation").

To implement that condition, all I need to do is add the Rx filter function to my pipeline. The filter function works much like a LINQ Where clause: It must be passed by another function (the selector function) that contains a test and returns a Boolean value based on data associated with the latest event. Only those events that pass the test will be passed to the subscribed observer. The filter function will automatically be handed the latest event object in the sequence so, in my selector function's test, I can use the event object's `target` property to retrieve the current value of the textbox and then check that value's length.

One more change to my pipeline: I'll remove the debounce function because it's hard to see what advantage it's giving me in this new UI. Still, after a significant change to my UI's interactions, the revised

code that creates my observable and subscribes my observers is still structurally identical to my previous version:

```
var getCustObsvble = Rx.DOM.keyup(getCustId)
    .filter(e => e.target.value.length == 4)
    .flatMapLatest(getCustomer);
getCustObsvble.subscribe(custObsrvr);
```

And, of course, I don't have to make any changes to my `custObsrvr` functions at all: They're isolated from my UI changes.

I can make one more optional change, this time to my transformation function. My transformation function has always been passed through three parameters—I've just been ignoring them when my event source was a button. The first parameter passed to the `flatMapLatest` transformation function is the event object for the event being observed. I can take advantage of that event object to eliminate the jQuery code that retrieved the customer Id and, instead, retrieve the value of the textbox from the event object. This change makes my code more loosely coupled because my transformation function is no longer tied to a specific element on the page.

My new transformation function looks like this:

```
function getCustomer(e)
{
    return Rx.DOM.getJSON("api/customerorders/customerbyid/" + e.target.value);
}
```

This is the beauty of the Rx abstraction: Changing from a single-event-producing element to a sequence-of-events-producing element has only required tinkering with the single line of code that makes up my pipeline (and gave me an opportunity to enhance my transformation function). Of all the changes I've made, the one most likely to cause me problems is renaming the variable that holds my input element. This change also emphasizes that, like LINQ, the secret to success with Rx is becoming familiar with the available operators.

Abstracting Web Service Calls

While the Rx abstraction makes changes to the UI look very much alike, a good abstraction should also do the same to the back-end processing. For example, what if the page is revised so that I retrieve not only customer data, but also the customer's sales orders? To make this interesting, I'll assume there's no navigation property that will let me retrieve the sales orders as part of the customer object and I have to make a second call.

With Rx, I first need to create a second transformation function that converts a customer Id into a collection of customer orders:

```
function getCustomerOrders(e) {
    return Rx.DOM.getJSON("customerorders/ordersbycustomerid/" + e.target.value);
}
```

Then I need to create a second observable that uses this transformation (this code looks very much like the code that creates my customer observable):

```
var getOrdersObsvble = Rx.DOM.keyup(getCustId)
    .filter(e => e.target.value.length == 4)
    .flatMapLatest(getCustomerOrders);
```

At this point, you might expect that combining and coordinating these observables could require a considerable amount of work. But, because Rx makes all observables look very much alike, you can combine the output from multiple observables into a single sequence that can be handled by a single (and relatively simple) observer.

For example, if I had several observables retrieving orders from multiple sources, I could use Rx's `merge` function to join all of the

order into a single sequence to which I could subscribe a single observer. The following code, for example, brings together observables retrieving current orders (`getCurrentOrdersObsvble`) and posted orders (`getPostedOrdersObsvble`); it then passes the resulting sequence to a single observer called `allOrdersObsrvr`:

```
Rx.Observable.merge(getCurrentOrdersObsvble, getPostedOrdersObsvble)
    .subscribe(allOrdersObsrvr);
```

In my case, I want to do something more interesting: combine an observable retrieving a customer object with an observable retrieving all the orders for that customer. Fortunately, RxJS has an operator for that: `combineLatest`. The `combineLatest` operator accepts two observables and a processing function. The processing function you pass to `combineLatest` is passed the latest result from each sequence and lets you specify how the results should be combined. In my case, `combineLatest` is providing more functionality than I need because I have only one result from each observable: the customer object from one observable and all of the customer's orders from the other.

In the following code, I add a new property (called `Orders`) to the customer object from my first observable and then put the result from the second observable into that property. Finally, I subscribe an observer called `CustOrdersObsrvr` to handle my new `Customer+Orders` object:

```
Rx.Observable.combineLatest(getCustObsvble, getOrdersObsvble,
    (c, ords) => {c.Orders = ords; return c;})
    .subscribe(CustOrdersObsrvr);
```

My `CustOrdersObsrvr` can now work with the new object I've created:

```
var custOrdersObsrvr = {
    onNext: co => {
        // ...Code to update the page with customer and orders data...
    },
    onError: err => {$("#Message").text("Unable to retrieve Customer and Orders: " + err.description)}
};
```

Wrapping Up

By abstracting the components of an application to just two kinds of objects (observables and observers) and providing a rich set of operators for managing and transforming the results from observables, RxJS provides a highly flexible (and maintainable) way for creating Web applications.

Of course, there's always a danger with using a powerful library that abstracts complex processes into simple code: When your application does something you didn't expect, debugging a problem can be a nightmare because you can't see the individual lines of code that the abstraction has eliminated. Of course, you no longer have to write all that code, either, and (presumably) the abstraction layer will have fewer bugs than the code you would've written. It's the usual trade-off between power and visibility.

By deciding for RxJS, the net benefit to you is that you can make significant changes to your application without having to make significant changes to your code. ■

PETER VOGEL is a systems architect and principal in PH&V Information Services. PH&V provides full-stack consulting from UX design through object modeling and database design. You can contact him at peter.vogel@phvis.com.

THANKS to the following Microsoft technical experts for reviewing this article: Stephen Cleary, James McCaffrey and Dave Sexton

DEVELOPED FOR INTUITIVE USE

DynamicPDF—Comprehensive PDF Solutions for .NET Developers

ceTe Software's DynamicPDF products provide real-time PDF generation, manipulation, conversion, printing, viewing, and much more. Providing the best of both worlds, the object models are extremely flexible but still supply the rich features you need as a developer. Reliable and efficient, the high-performance software is easy to learn and use. If you do encounter a question with any of our components, simply contact ceTe Software's readily available, industry-leading support team.



DynamicPDF

[WWW.DYNAMICPDF.COM](http://www.DynamicPDF.com)



TRY OUR PDF SOLUTIONS FREE TODAY!

www.DynamicPDF.com/eval or call 800.631.5006 | +1 410.772.8620

ceTe software

Cross-Platform Productivity with Xamarin

Kevin Ashley

If you're in Silicon Valley these days, you'll notice that investors and developers alike are mostly conservative about the technologies they use. Most would hire an iOS developer first, followed (if they had enough funding) by an Android and/or a Windows developer. This approach is extremely inefficient, however, and leads to multiple code rewrites: iOS developers use Objective-C; Android developers use Java; Windows developers use C#. They rarely talk to each other, much less participate in code sharing, and this results in apps that are inconsistent across platforms—and usually millions of dollars wasted to support completely separate development branches and technologies. For an effective and consistent development strategy, using a cross-platform technology like Xamarin is the key to productivity. And with the rise of native ahead-of-time (AOT) compilation technologies that translate natively from high-level languages like C# to static code for native platforms, the programming model where separate Android, iOS and Windows developers are needed is becoming a thing of the past.

This is a new direction for traditional apps, but it's already been successful in other areas, such as gaming. For example, more than

half of all 3D games for iOS, Android and Windows are developed with Unity. When you build games in Unity, you mostly develop in C# on a subset of Mono, and that's not unlike using C# with Xamarin. Unity games run on 20-plus platforms, including Windows, Android, iOS, OS X, Xbox, PlayStation and more. This is a great path to follow for non-gaming apps, and Xamarin provides an excellent platform for doing that.

Can the same approach be used for other apps, not just games? With Xamarin, it can: I used it to create Active Fitness, a cloud-based fitness platform and mobile apps with more than 2 million users around the world (activefitness.co). I started using Xamarin early in the development cycle, beginning with very early versions, and was able to create a cross-platform solution for Windows, iOS and Android, based on the Microsoft Azure cloud. Active Fitness, shown in **Figure 1**, boasts upward of 90 percent code sharing across its native iOS, Windows and Android apps. That in itself is a productivity gain. In addition, most of Xamarin, including Xamarin.Forms (bit.ly/2a8Yo4g), is open source; the platform enjoys great community support; and the performance is identical to native code, thanks to AOT support on embedded systems. Moreover, Xamarin is included in Visual Studio at no additional cost, including the free Community Edition.

This article discusses:

- Design aspects of cross-platform apps with Xamarin
- Device-specific code and Xamarin
- Accessing the cloud from Xamarin
- Extensions, components and plug-ins
- Support for wearables
- Compiled XAML

Technologies discussed:

Xamarin, Android, iOS and Windows

Design Aspects of Cross-Platform Apps with Xamarin

A few years back, mobile app design was strikingly different on each platform. An iOS app, for example, was really not at all like an Android or Windows app in design. That's no longer the case. Today, most modern apps for Windows, Android, and iOS use similar concepts and design paradigms. What happened? Microsoft created the Modern design language used by Universal Windows

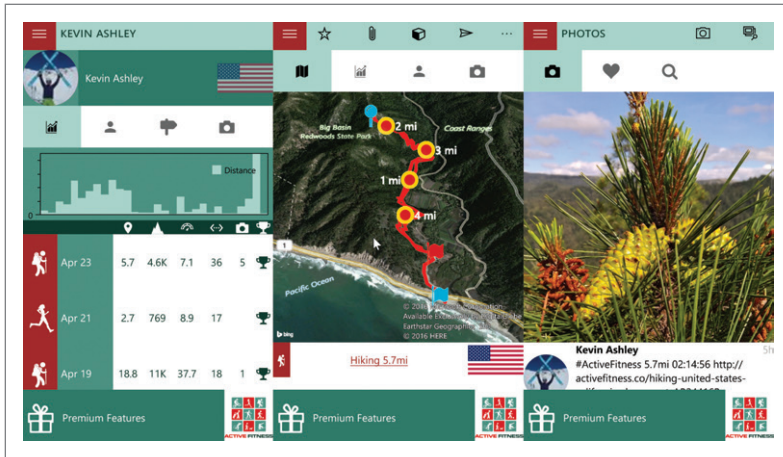


Figure 1 Active Fitness in Action, Demonstrating 3D Maps and a Rich UI

Platform (UWP) apps; Google came up with Material design; and Apple introduced a more modern-looking OS design. All these changes made apps in iOS, Android, and Windows look and act more alike, simplifying developers' work (see **Figure 2**). Developers even flock to iconic fonts, such as Font Awesome, that use similar iconic concepts for actions across multiple platforms. Platforms still have visual differences, of course; for example, Windows includes live tiles. Perhaps surprisingly, though, if you build a cross-platform app, those are easy to add once the core is there.

When developing with Xamarin you have the ability to craft a beautiful UI for each platform directly from inside Visual Studio. This means using iOS Storyboards, Android XML, and Windows XAML, with access to every control or widget and a huge number of custom controls from vendors. Moreover, Xamarin delivers 100 percent API access completely in C# for each platform.

With the Xamarin.Forms library, which provides an API abstraction for building a shared UI, many design concepts are automatically translated to native controls for each platform via Xamarin custom

renderers, giving you the ability to customize a cross-platform control to its native representation on each platform. So you get the productivity advantage of a truly cross-platform app, plus unlimited customization possibilities!

One huge benefit of using Xamarin.Forms is that it's simply an add-on, a library you add to your project. In terms of productivity and the lifecycle management of your app, this is priceless because you can create code that doesn't depend much on platform release cycles, which are usually much slower. Xamarin has always had same-day compatibility with major releases of iOS, Android, and Windows, and now that it's open source, if you need to add a control to Xamarin.Forms, you can always do it yourself by looking at the source code or participating in the GitHub repository for Xamarin.Forms.

In addition, Xamarin offers virtually unlimited extensibility options via the custom control renderers for native platforms. If you already have a XAML app, you're in luck because Xamarin.Forms is just another dialect of XAML, and if your app already uses Silverlight, UWP, or XAML, you can easily add a cross-platform companion in iOS or Android.

Creating Master-Detail Navigation with Xamarin

Let's take a look at the advantage of using Xamarin by focusing on one example: a typical app in iOS, Android and Windows containing a master-detail (or "hamburger") navigation pattern, from the open source Xamarin samples repository (bit.ly/29Tk9VJ). If you were to write this app as traditional iOS, Android, and Windows developers do, you'd end up with three separate projects in Objective-C, Java, and C#, and you'd have to use different controls for each: SplitView in Windows, and similar patterns in iOS and Android. Instead, you can use a single Xamarin.Forms instance to do all that hard work for all platforms, as **Figure 3** shows.

In the App.cs file of the portable project, I instantiate the root page of the app:

```
public App ()
{
    MainPage =
        new MasterDetailPageNavigation.
        MainPage ();
}
```

Note that most of the project code resides in a portable project, shared among all OSes, plus in Xamarin.Forms you can use XAML. The XAML in **Figure 4** likely looks pretty simple if you're familiar with the UWP native SplitView control.

Xamarin provides a lot of flexibility in terms of what developers can do with conditional compilation and platform selection. The following code changes the icon of the master page if the platform is Windows:

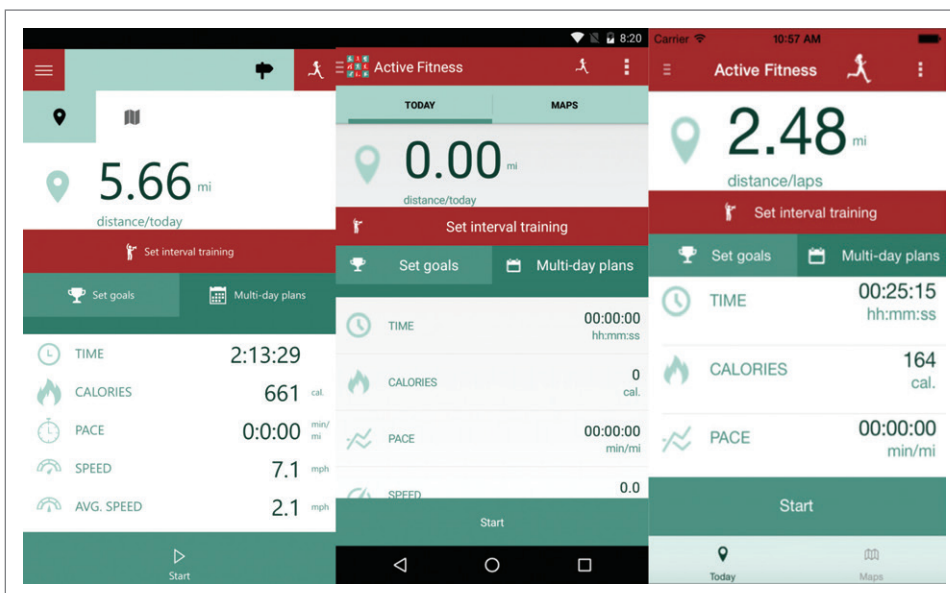


Figure 2 Active Fitness on Windows (Left), Android (Middle) and iOS (Right)

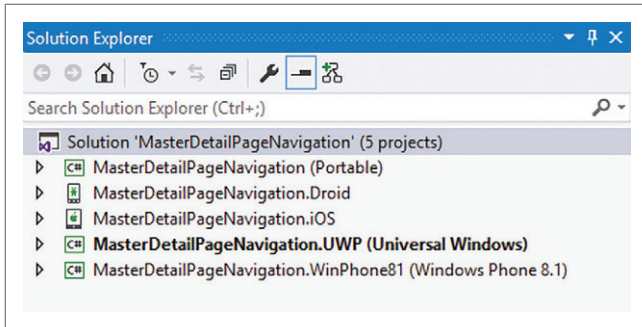


Figure 3 Xamarin Master-Detail Project with Android, iOS and Windows Targets

```
if (Device.OS == TargetPlatform.Windows)
{
    Master.Icon = "hamburger.png";
}
```

Device-Specific Code and Xamarin

So far so good, but can I really get very close to the physical platform in my Xamarin code? In other words, can I differentiate code created for different types of devices? Fortunately, Xamarin is one of the few cross-platform technologies that support native coding and API access. Similar to UWP apps that let you differentiate device family (phone, tablet or desktop), Xamarin contains a very useful `Device.Idiom` selector, as **Figure 5** shows.

Figure 6 shows another very useful selector, `Device.OnPlatform`, which can be used in both the codebehind and in your XAML to execute different actions depending on the target OS.

In XAML, Xamarin adds `OnPlatform` tags supporting different platforms, for example:

```
<Button Text="Start Timer"
        Clicked="TimerClicked"
        BackgroundColor="Gray"
        HorizontalOptions="Center">
    <OnPlatform x:TypeArguments="x:Double"
        iOS="200"
        Android="300"
        WinPhone="100" />
</Button.WidthRequest>
</Button>
```

But what if I want to customize a control provided by Xamarin.Forms on a native platform? I mentioned earlier the Xamarin feature called custom renderers, which provides that flexibility. Custom renderers are classes you can derive from built-in objects and provide functionality in platform-specific implementations. Let's say I want to build my own `CustomMap` control:

```
public class CustomMap : Map
{
    public List<CustomPin> CustomPins { get; set; }
}
```

On a native platform, the signature of my object will include the following:

```
[assembly:ExportRenderer (typeof(CustomMap), typeof(CustomMapRenderer))]
```

So, in Android, iOS and Windows, I can always customize the way my Map control looks and feels. Interestingly, the wrapper for the Xamarin Map control natively wraps the Windows Maps control on Windows, the Google Maps control on Android and the native iOS Maps control on iOS. What else can you customize with custom renderers? Pretty much any control! The Map control

is a fairly sophisticated example, but you can also customize buttons, labels, sliders or any control available in Xamarin.Forms.

With Xamarin.Forms 2.2, Xamarin now includes native embedding, which allows adding any native control to a Xamarin.Forms app (bit.ly/29IEvXH).

Accessing the Cloud from Xamarin

Azure has always been cross-platform-friendly, with a variety of SDKs for many different platforms, including iOS, Android, and dedicated cross-platform examples and support for Xamarin. My Active

Figure 4 Creating a Master-Detail Page Using XAML

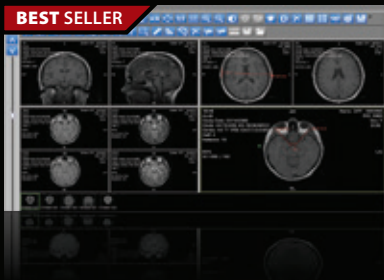
```
<?xml version="1.0" encoding="UTF-8"?>
<MasterDetailPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    xmlns:local="clr-namespace:MasterDetailPageNavigation;assembly=MasterDetailPageNavigation"
    x:Class="MasterDetailPageNavigation.MainPage">
    <MasterDetailPage.Master>
        <local:MasterPage x:Name="masterPage" />
    </MasterDetailPage.Master>
    <MasterDetailPage.Detail>
        <NavigationPage>
            <x:Arguments>
                <local:ContactsPage />
            </x:Arguments>
        </NavigationPage>
    </MasterDetailPage.Detail>
</MasterDetailPage>
```

Figure 5 Using `Device.Idiom` to Select the Platform

```
switch (Device.Idiom)
{
    case TargetIdiom.Phone:
        heading.Text += " Phone ";
        break;
    case TargetIdiom.Tablet:
        heading.Text += " Tablet ";
        break;
    case TargetIdiom.Desktop:
        heading.Text += " Desktop ";
        break;
    default:
        heading.Text += " unknown ";
        break;
}
```

Figure 6 Using `Device.OnPlatform` to Target Different Platforms

```
// Device.OnPlatform (Action)
//
var box = new BoxView {
    Color = Color.Green,
    WidthRequest = Device.OnPlatform(30, 40, 50),
    HorizontalOptions = LayoutOptions.Center
};
Device.OnPlatform(
    iOS: () =>{
        box.Color = box.Color.MultiplyAlpha(0.5);
        heading.TextColor = Color.Blue;
    },
    Android: () =>{
        box.Color = box.Color.AddLuminosity(0.3);
        heading.TextColor = Color.FromRgb(115, 129, 130);
    },
    WinPhone: () =>{
        box.Color = box.Color.AddLuminosity(0.3);
        heading.TextColor = Color.Accent;
    },
    Default: () =>{
        heading.Text = "what platform is this?!" + Device.OS;
    }
);
```



LEADTOOLS Medical Imaging SDKs V19

from \$4,995.00 SRP



Powerful DICOM, PACS, and HL7 functionality.

- Load, save, edit, annotate & display DICOM Data Sets with support for the latest specifications
- High-level PACS Client and Server components and frameworks
- OEM-ready HTML5 Zero-footprint Viewer and DICOM Storage Server apps with source code
- Medical-specific image processing functions for enhancing 16-bit grayscale images
- Native libraries for .NET, C/C++, HTML5, JavaScript, WinRT, iOS, OS X, Android, Linux, & more



DevExpress DXperience 16.1

from \$1,439.99



The complete range of DevExpress .NET controls and libraries for all major Microsoft platforms.

- WinForms - New TreeMap control, Chart series types and Unbound Data Source
- WPF - New Wizard control and Data Grid scrollbar annotations
- ASP.NET - New Vertical Grid control, additional Themes, Rich Editor Spell Checking and more
- Windows 10 Apps - New Hamburger Sub Menus, Splash Screen and Context Toolbar controls
- CodeRush - New debug visualizer expression map and code analysis diagnostics



Help & Manual Professional

from \$586.04



Help and documentation for .NET and mobile applications.

- Powerful features in an easy, accessible and intuitive user interface
- As easy to use as a word processor, but with all the power of a true WYSIWYG XML editor
- Single source, multi-channel publishing with conditional and customized output features
- Output to responsive HTML, CHM, PDF, MS Word, ePub, Kindle or print
- Styles and Templates give you full design control



Aspose.Total for .NET

from \$2,939.02



Every Aspose .NET component in one package.

- Programmatically manage popular file formats including Word, Excel, PowerPoint and PDF
- Work with charts, diagrams, images, project plans, emails, barcodes, OCR and OneNote files alongside many more document management features in .NET applications
- Common uses also include mail merging, adding barcodes to documents, building dynamic reports on the fly and extracting text from most document types

Fitness app was able to scale to 2 million users because of this. Thanks to Azure—its reliability, support and performance—users are able to keep running, jogging and performing various sports activities every day with thousands of devices connected every minute. You can find several great guides dedicated to Xamarin on the Azure Web site, which can help you discover the Microsoft cloud services most suitable for your apps (bit.ly/2a5kciF). In the meantime, I'd like to mention a few common areas where the cloud is very useful for mobile apps:

Authentication: If you need to authenticate or identify users, you'll likely need to use cloud services in your Xamarin app. Visit bit.ly/29HIDD3 to see how.

Adding push notifications to your apps: Push notifications not only provide a communication mechanism for apps, but also help update platform-specific features, such as tiles. Find out more at bit.ly/29HIDD3.

Offline synchronization: Mobile apps in everyday use are not always online, yet the data needs to synchronize seamlessly, even when devices become disconnected from the back end. See my *MSDN Magazine* article about occasionally disconnected data sets at msdn.com/magazine/dn890372, as well as a quick start for Xamarin apps at bit.ly/29QkXqT.

Azure Mobile Apps is a great starting point for integrating many cloud services including storage, notifications and authentication. Check out the tutorial at bit.ly/29K3IHl, which is dedicated to Xamarin apps.

Extensions, Components and Plug-ins

Xamarin succeeded and evolved via considerable support of components, plug-ins and extensions. Many extensions are available on GitHub or via NuGet. Some libraries—for example, Android Support Libraries, Google Play Services client libraries, Facebook SDKs, and event charting libraries such as OxyPlot—are supported by Xamarin or its community. Plug-ins are unique offerings that enable developers to access native functionality from shared code. This means that if you want to access GPS functionality you no longer have to write the code three times—you simply download a NuGet package and access the GPS right in your shared code. Some plug-ins are open source and are supported by developers. For example, James Montemagno has developed a large number of plug-ins for Xamarin, such as Connectivity, Settings, Media and others (bit.ly/2a2mM7J).

Figure 7 shows several useful extensions I recommend using as you get deeper into Xamarin development.

For example, checking connectivity with the Connectivity plug-in in a cross-platform solution becomes a one-liner instead of platform-specific code that needs to be implemented separately in iOS, Android and Windows:

```
public static bool IsOnline
{
    get
    {
        return Plugin.Connectivity.CrossConnectivity.Current.IsConnected;
    }
}
```

Figure 7 Xamarin Extensions

Name	Description	NuGet Link	Docs and Source on GitHub
Battery Status	Gather battery level, charging status and type.	bit.ly/2a4gbZ6	bit.ly/2a5Ofqm
Barcode Scanner	Scan and create barcodes with ZXing.NET.Mobile.	bit.ly/2a5Ofqm	bit.ly/29QyKY9
Compass	Access device compass heading.	bit.ly/2a32UAZ	bit.ly/29KfcKV
Connectivity	Get network connectivity info such as type and if connection is available.	bit.ly/29QDpLO	bit.ly/2a33PBr
Cryptography	PCL Crypto provides a consistent, portable set of crypto APIs.	bit.ly/29Qz5AE	bit.ly/29PzwAb
Device Info	Get properties about a device, such as OS, model, version and Id.	bit.ly/29PzPeg	bit.ly/29QzSBq
Device Motion	Provides access to Accelerometer, Gyroscope, Magnetometer and Compass.	bit.ly/2a6SzTk	bit.ly/2a35maG
Embedded Resource	Unpack embedded resource for cross-platform use.	bit.ly/29J6Wf3	bit.ly/29J6z46
External Maps	Launch external maps from latitude/longitude or address.	bit.ly/29KgNR0	bit.ly/2abuJbl
File System	PCL Storage offers cross-platform storage APIs.	bit.ly/29LEOru	bit.ly/28Ju1AB
Geolocator	Detect GPS location of device.	bit.ly/2a70ekG	bit.ly/29Tpuvd
Local Notifications	Show local notifications.	bit.ly/2arOGYf	bit.ly/29TpSd7
Media	Take or pick photos and videos.	bit.ly/2a6rpxi	bit.ly/29TqlMm
Messaging	Make phone call, send sms and send e-mail.	bit.ly/2a8Uie5	bit.ly/29SEdDm
Permissions	Check and request runtime permissions.	bit.ly/29Tnvo8	bit.ly/29S6ZKv
Akavache key-value store	An asynchronous, persistent (writes to disk) key-value store.	bit.ly/29Tou7l	bit.ly/2aRPSLf
Push Notifications	Cross-platform push notifications for iOS and Android.	bit.ly/29ToDs5	bit.ly/2aex4CR
Settings	Simple, consistent cross-platform settings API.	bit.ly/29ToTXT	bit.ly/2a6tn0Q
Share	Easily share text, links or open a browser.	bit.ly/2aa2R51	bit.ly/2aa3sUk
Sockets	TCP and UDP listeners and clients, plus UDP multicast.	bit.ly/1rQlyyR	bit.ly/1y1UHPb
Text to Speech	Text to speech from shared code.	bit.ly/29S7Yud	bit.ly/29MMA3S
User Dialogs	Enables messagebox-style dialogs.	bit.ly/2aa4dMV	bit.ly/29Tqzkd
Version Tracking	Track which versions of your app a user has installed.	bit.ly/29S8YhH	bit.ly/2a74INW

Why Fortune 500 companies choose **RavenDB**?

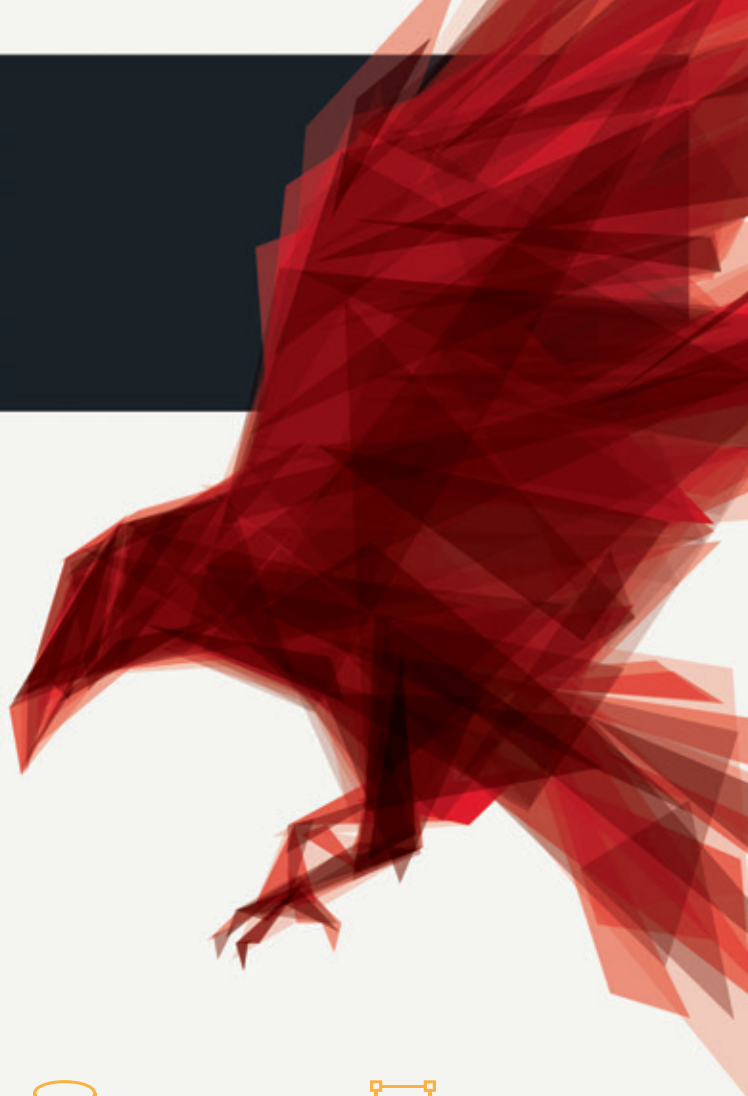
In a world where data is one of the most important assets of any business the database technology should not only be protecting its data but also enhancing its business.

To address both of those needs, Hibernating Rhinos has introduced its NoSQL database called RavenDB and for the past few years, due to enhanced capabilities, it has become the choice of Fortune 500 companies.

The protection of data comes with meeting all the ACID parameters, being fully transactional and having extended failover support to guarantee you that the data will be safe and sound even when node failure happens. Moreover, the extended replication features allow businesses to setup complex failover clusters to move their protection to the next level and ensure availability or enhance their work by enabling sophisticated sharding and load balancing capabilities.

The out of the box querying features, high-performance and self-optimization assure that the database will not stand in the way of company growth.

All this is provided with user-friendly HTML5 management interface, ease of deployment and top-notch C# and Java client libraries.



Schema-free



Scalable



RavenFS



Easy to use



Transactional



High Performance



Extensible



Designed with Care

NEW



Monitoring



Hot Spare



Clustering

RAVENDB 3.5
RELEASED

ravendb.net

You'll find even more plug-ins and extensions are available at bit.ly/29XZ3VM.

Device Support: Android Wear, WatchKit and Microsoft Band

Xamarin has always been very responsive to all sorts of devices and device platforms on the market. I had a pleasure of working with the Microsoft Band team, and at Build 2016 we jointly announced some pretty exciting functionality for Active Fitness that added support for more than 50 fitness activities, including skiing, snowboarding and more. Additionally, Active Fitness now supports Android Wear, as the beautiful watch face in **Figure 8** shows. If you install Active Fitness on your Android device, the watch face is fully supported.

With Xamarin, you can add apps and support to gadgets and devices using the following:

- Android Wear: Xamarin includes code examples and support, including creating watch faces in C#.
- Apple WatchKit: For Xamarin support, check the documentation at bit.ly/29XZ3VM.
- Microsoft Band: In addition to the Microsoft Band SDK, there's a fantastic Microsoft Band wrapper for cross-platform Xamarin apps available on GitHub (bit.ly/29WeDli) and NuGet (bit.ly/29S0oLA).

Performance and Compiled XAML

Xamarin.Forms uses a combination of XAML and codebehind logic. If you prefer, you can use only code to instantiate and manage your UI objects. In fact, nothing prevents you from coding most of your UI, apart from efficiency (the declarative nature of XAML makes it very efficient to write and manage). But if you really want to, you can derive your Page from ContentPage and write all the logic in code, no XAML involved, as **Figure 9** shows.

In this case, all your UI code is really compiled C#. Compiled XAML uses a similar idea: You can tell Xamarin to precompile all your XAML, which makes UI code contained in the XAML files much faster to load and execute. There are several benefits with compiled XAML: immediate compile-time checking, removing

Figure 9 Deriving a Page from ContentPage Without XAML

```
class ButtonCodePage : ContentPage
{
    int count = 0;

    public ButtonCodePage()
    {
        Button button = new Button
        {
            Text = String.Format("Tap for click count!")
        };
        button.Clicked += (sender, args) =>
        {
            count++;
            button.Text =
                String.Format("{0} click{1}!", count, count == 1 ? "" : "s");
        };

        this.Content = button;
    }
}
```

load and instantiation time, and reducing the size of your executables. You can enable XAML compilation at assembly:

```
[assembly: XamlCompilation(XamlCompilationOptions.Compile)]
namespace MyApp
```

or at the class level:

```
[XamlCompilation(XamlCompilationOptions.Compile)]
public partial class MyPage
```

Wrapping Up

Developing with Xamarin, I learned to focus on productivity and to harness the power of the Microsoft open source community. As I've noted, Xamarin has been very responsive to open source and device trends: Android Wear, Apple Watch, Microsoft Band, and the latest versions of Android, iOS and Windows. Here are a few notable new additions to Xamarin that make it a truly exciting platform:

- Data Pages: Simplifies data binding and makes complex multi-page apps even easier to build (bit.ly/29SRlLk).
- Localization and Multilingual App Toolkit: Fully supports Xamarin projects, adding industry-standard XLIFF support to your projects (bit.ly/2a5Uzwx).
- Native Control Embedding: Makes adding native controls to Xamarin apps super easy (bit.ly/29IEvxH).
- Effects: Adds platform-specific effects to Xamarin apps (bit.ly/29DRbD).

Xamarin provides a powerful and efficient way to build cross-platform apps of literally any complexity that work in iOS, Android and Windows. You can truly focus on app functionality and content and build a powerful app that'll work in any OS. And once you start building your first Xamarin app, you'll soon discover a large and vibrant community of cross-platform developers. ■

KEVIN ASHLEY is an architect evangelist for Microsoft. He's coauthor of "Professional Windows 8 Programming" (Wrox, 2012) and a developer of top apps and games, most notably Active Fitness (activefitness.co). He often presents on technology at various events, industry shows and Webcasts. In his role, he works with startups and partners, advising on software design, business and technology strategy, architecture, and development. Follow his blog at kevinashley.com and on Twitter: @kashleytwit.

THANKS to the following Microsoft technical experts for reviewing this article: James Montemagno

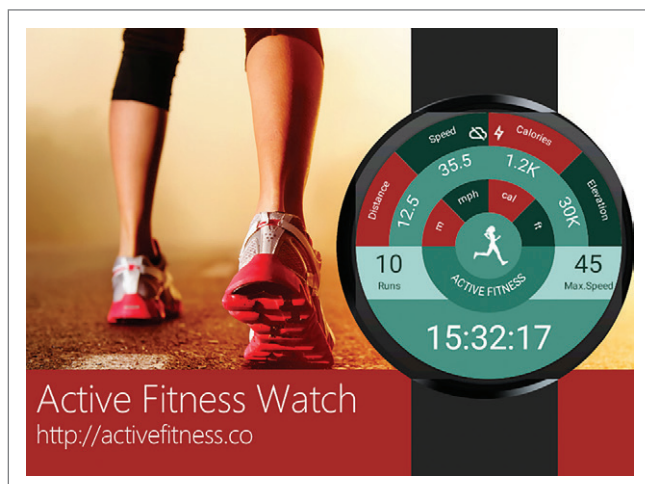


Figure 8 Android Wear Watch Face for Active Fitness App

We Made WPF GREAT!



Xceed Business Suite for WPF

- 104 high-quality WPF controls and themes
- The fastest and most advanced datagrid
- Widely used controls, over 500K downloads
- Created by WPF pioneers
- 10 years of WPF excellence
- Frequently updated and improved



Unicode Encoding Conversions with STL Strings and Win32 APIs

Giovanni Dicanio

Unicode is the de facto standard for representing international text in modern software. According to the official Unicode consortium's Web site (bit.ly/1Rtdulx), "Unicode provides a unique number for every character, no matter what the platform, no matter what the program, no matter what the language." Each of these unique numbers is called a code point, and is typically represented using the "U+" prefix, followed by the unique number written in hexadecimal form. For example, the code point associated to the character "C" is U+0043. Note that Unicode is an industry standard that covers most of the world's writing systems, including ideographs. So, for example, the Japanese *kanji* ideograph 学, which has "learning" and "knowledge" among its meanings, is

associated to the code point U+5B66. Currently, the Unicode standard defines more than 1,114,000 code points.

From Abstract Code Points to Actual Bits: UTF-8 and UTF-16 Encodings

A code point is an abstract concept, though. For a programmer, the question is: How are these Unicode code points represented concretely using computer bits? The answer to this question leads directly to the concept of Unicode encoding. Basically, a Unicode encoding is a particular, well-defined way of representing Unicode code point values in bits. The Unicode standard defines several encodings, but the most important ones are UTF-8 and UTF-16, both of which are variable-length encodings capable of encoding *all* possible Unicode "characters" or, better, code points. Therefore, conversions between these two encodings are lossless: No Unicode character will be lost during the process.

UTF-8, as its name suggests, uses 8-bit code units. It was designed with two important characteristics in mind. First, it's backward-compatible with ASCII; this means that each valid ASCII character code has the same byte value when encoded using UTF-8. In other words, valid ASCII text is automatically valid UTF-8-encoded text.

Second, because Unicode text encoded in UTF-8 is just a sequence of 8-bit byte units, there's no *endianness* complication. The UTF-8 encoding (unlike UTF-16) is endian-neutral by design. This is an important feature when exchanging text across different computing systems that can have different hardware architectures with different endianness.

This article discusses:

- UTF-8 and UTF-16 encodings
- Converting between UTF-8 and UTF-16 using Win32 APIs
- Wrapping `MultiByteToWideChar` in a modern-style C++ function for UTF-8 to UTF-16 conversions
- Handling errors with C++ exceptions

Technologies discussed:

C++ Unicode, Win32 APIs, STL strings

Code download available at:

msdn.com/magazine/0916magcode



R#

—

ReSharper

THE LEGENDARY EXTENSION
TO VISUAL STUDIO*

jetbrains.com/msdn

JET
BRAINS

—

THE DRIVE
TO DEVELOP

* WARNING! PROLONGED USE
MAY CAUSE ADDICTION



Considering the two Unicode characters I mentioned before, the capital letter C (code point U+0043) is encoded in UTF-8 using the single byte 0x43 (43 hexadecimal), which is exactly the ASCII code associated with the character C (as per the UTF-8 backward compatibility with ASCII). In contrast, the Japanese ideograph 学 (code point U+5B66) is encoded in UTF-8 as the three-byte sequence 0xE5 0xAD 0xA6.

UTF-8 is the most-used Unicode encoding on the Internet. According to recent W3Techs statistics available at bit.ly/1UT5EBC, UTF-8 is used by 87 percent of all the Web sites it analyzed.

UTF-16 is basically the de facto standard encoding used by Windows Unicode-enabled APIs. UTF-16 is the “native” Unicode encoding in many other software systems, as well. For example, Qt, Java and the International Components for Unicode (ICU) library, just to name a few, use UTF-16 encoding to store Unicode strings.

UTF-16 uses 16-bit code units. Just like UTF-8, UTF-16 can encode *all* possible Unicode code points. However, while UTF-8 encodes each valid Unicode code point using one to four 8-bit byte units, UTF-16 is, in a way, simpler. In fact, Unicode code points are encoded in UTF-16 using just one or two 16-bit code units. However, having code units larger than a single byte implies endianness complications: in fact, there are both *big-endian* UTF-16 and *little-endian* UTF-16 (while there’s just one endian-neutral UTF-8 encoding).

Unicode defines a concept of plane as a continuous group of 65,536 (2¹⁶) code points. The first plane is identified as plane 0 or Basic Multilingual Plane (BMP). Characters for almost all modern languages and many symbols are located in the BMP, and all these BMP characters are represented in UTF-16 using a single 16-bit code unit.

Supplementary characters are located in planes other than the BMP; they include pictographic symbols like Emoji and historic scripts like Egyptian hieroglyphs. These supplementary characters outside the BMP are encoded in UTF-16 using two 16-bit code units, also known as surrogate pairs.

The capital letter C (U+0043) is encoded in UTF-16 as a single 16-bit code unit 0x0043. The ideograph 学 (U+5B66) is encoded in UTF-16 as the single 16-bit code unit 0x5B66. For many Unicode characters there’s an immediate, direct correspondence between their code point “abstract” representation (such as U+5B66) and their associated UTF-16 encoding in hex (for example, the 0x5B66 16-bit word).

To have a little fun, let’s take a look at some pictographic symbols. The Unicode character “snowman” (☺, U+2603) is encoded in UTF-8 as the three-byte sequence: 0xE2 0x98 0x83; however, its UTF-16 encoding is the single 16-bit unit 0x2603. The Unicode character “beer mug” (🍺, U+1F37A), which is located outside the BMP, is encoded in UTF-8 by the four-byte sequence 0xF0 0x9F 0x8D 0xBA. Its UTF-16 encoding instead uses two 16-bit code units, 0xD83C 0xDF7A, an example of a UTF-16 surrogate pair.

Converting Between UTF-8 and UTF-16 Using Win32 APIs

As discussed in the previous paragraphs, Unicode text is represented inside a computer’s memory using different bits, based on the particular Unicode encoding. Which encoding should you use? There isn’t a single answer to this question.

Recently, conventional wisdom seems to suggest that a good approach consists of storing Unicode text encoded in UTF-8 in

instances of the `std::string` class in cross-platform C++ code. Moreover, there’s general agreement that UTF-8 is the encoding of choice for exchanging text across application boundaries and across different machines. The fact that UTF-8 is an endian-neutral format plays an important role in this. In any case, conversions between UTF-8 and UTF-16 are required at least at the Win32 API boundary, because Windows Unicode-enabled APIs use UTF-16 as their native encoding.

Now let’s dive into some C++ code to implement these Unicode UTF-8/UTF-16 encoding conversions. There are two key Win32 APIs that can be used for this purpose: `MultiByteToWideChar` and its symmetric `WideCharToMultiByte`. The former can be invoked to convert from UTF-8 (“multi-byte” string in the specific API terminology) to UTF-16 (“wide char” string); the latter can be used for the opposite. Because these Win32 functions have similar interfaces and usage patterns, I’ll focus only on `MultiByteToWideChar` in this article, but I have included C++-compatible code that uses the other API as part of this article’s download.

Using Standard STL String Classes to Store Unicode Text

Because this is a C++ article, there’s a valid expectation of storing Unicode text in some sort of string class. So, the question now becomes: What kind of C++ string classes can be used to store Unicode text? The answer is based on the particular encoding used for the Unicode text. If UTF-8 encoding is used, because it’s based on 8-bit code units, a simple `char` can be used to represent each of these code units in C++. In this case the STL `std::string` class, which is char-based, is a good option to store UTF-8-encoded Unicode text.

On the other hand, if the Unicode text is encoded in UTF-16, each code unit is represented by 16-bit words. In Visual C++, the `wchar_t` type is exactly 16 bits in size; consequently, the STL `std::wstring` class, which is `wchar_t`-based, works fine to store UTF-16 Unicode text.

It’s worth noting that the C++ standard doesn’t specify the size of the `wchar_t` type, so while it amounts to 16 bits with the Visual C++ compiler, other C++ compilers are free to use different sizes. And, in fact, the size of `wchar_t` defined by the GNU GCC C++ compiler on Linux is 32 bits. Because the `wchar_t` type has different sizes on different compilers and platforms, the `std::wstring` class, which is based on that type, is *non-portable*. In other words, `wstring` can be used to store Unicode text encoded in UTF-16 on Windows with the Visual C++ compiler (where the size of `wchar_t` is 16 bits), but not on Linux with the GCC C++ compiler, which defines a different-sized 32-bit `wchar_t` type.

There is actually another Unicode encoding, which is less well-known and less used in practice than its siblings: UTF-32. As its name clearly suggests, it’s based on 32-bit code units. So a GCC/Linux 32-bit `wchar_t` is a good candidate for the UTF-32 encoding on the Linux platform.

This ambiguity on the size of `wchar_t` determines a consequent lack of portability of C++ code based on it (including the `std::wstring` class itself). On the other hand, `std::string`, which is char-based, is portable. However, from a practical perspective, it’s worth noting that the use of `wstring` to store UTF-16 encoded text is just fine in Windows-specific C++ code. In fact, these portions of code already interact with Win32 APIs, which are, of course, platform-specific by definition. So adding `wstring` to the mix doesn’t change the situation.

Finally, it’s important to note that, because both UTF-8 and UTF-16 are variable-length encodings, the return values of the `string::length`

and `wstring::length` methods in general don't correspond to the number of Unicode characters (or code points) stored in the strings.

The Conversion Function Interface Let's develop a function to convert Unicode text encoded in UTF-8 to the equivalent text encoded using UTF-16. This can come in handy, for example, when you have some cross-platform C++ code that stores UTF-8-encoded Unicode strings using the STL `std::string` class, and you want to pass that text to Unicode-enabled Win32 APIs, which typically use the UTF-16 encoding. Because this code is talking to Win32 APIs, it's already non-portable, so `std::wstring` is well-suited to store UTF-16 text here. A possible function prototype is:

```
std::wstring Utf8ToUtf16(const std::string& utf8);
```

This conversion function takes as input a Unicode UTF-8-encoded string, which is stored in the standard STL `std::string` class. Because this is an input parameter, it's passed by const reference (`const &`) to the function. As the result of the conversion, a string encoded in UTF-16 is returned, stored in a `std::wstring` instance. However, during Unicode encoding conversions, things can go wrong. For example, the input UTF-8 string might contain an invalid UTF-8 sequence (which can be the result of a bug in other parts of the code, or it could end up there as a result of some malicious activity). In such cases, the best thing from a security perspective is to fail the conversion, instead of consuming potentially dangerous byte sequences. The conversion function can handle cases of invalid UTF-8 input sequences by throwing a C++ exception.

Defining an Exception Class for Conversion Errors What kind of C++ class can be used to throw an exception in case a Unicode encoding conversion fails? An option might be using a class already defined in the standard library, for example: `std::runtime_error`. However, I prefer defining a new custom C++ exception class for that purpose, deriving it from `std::runtime_error`. When Win32 APIs like `MultiByteToWideChar` fail, it's possible to call `GetLastError` to get further information about the cause of the failure. For example, in case of invalid UTF-8 sequences in the input string, a typical error code returned by `GetLastError` is `ERROR_NO_UNICODE_TRANSLATION`. It makes sense to add this piece of information to the custom C++ exception class; it can come in handy later for debugging purposes. The definition of this exception class can start like this:

```
// utf8except.h
#pragma once

#include <stdint.h> // for uint32_t
#include <stdexcept> // for std::runtime_error

// Represents an error during UTF-8 encoding conversions
class Utf8ConversionException
: public std::runtime_error
{
    // Error code from GetLastError()
    uint32_t _errorCode;
```

Note that the value returned by `GetLastError` is of type `DWORD`, which represents a 32-bit unsigned integer. However, `DWORD` is a Win32-specific non-portable typedef. Even if this C++ exception class is thrown from Win32-specific portions of C++ code, it can be caught by cross-platform C++ code! So, it does make sense to use portable typedefs instead of Win32-specific ones; `uint32_t` is an example of such types.

Next, a constructor can be defined, to initialize instances of this custom exception class with an error message and error code:

```
public:
    Utf8ConversionException(
        const char* message,
        uint32_t errorCode
    )
        : std::runtime_error(message)
        , _errorCode(errorCode)
    { }
```

Finally, a public getter can be defined to provide read-only access to the error code:

```
uint32_t ErrorCode() const
{
    return _errorCode;
}

}; // Exception class
```

Because this class is derived from `std::runtime_error`, it's possible to call the `what` method to get the error message passed in the constructor. Note that in the definition of this class, only portable standard elements have been used, so this class is perfectly consumable in cross-platform portions of C++ code, even those located far away from the Windows-specific throwing point.

Converting from UTF-8 to UTF-16: MultiByteToWideChar in Action

Now that the prototype of the conversion function has been defined and a custom C++ exception class implemented to properly represent UTF-8 conversion failures, it's time to develop the body of the conversion function. As already anticipated, the conversion work from UTF-8 to UTF-16 can be done using the `MultiByteToWideChar` Win32 API. The terms “multi-byte” and “wide-char” have roots in historical reasons. Basically, this API and its symmetric sibling `WideCharToMultiByte` were initially meant to convert between text stored in specific code pages and Unicode text, which uses the UTF-16 encoding in Win32 Unicode-enabled APIs. Wide char refers to `wchar_t`, so it's associated to a `wchar_t`-based string, which is a UTF-16-encoded string. In contrast, a multi-byte string is a sequence of bytes expressed in a code page. The legacy concept of code page was then extended to include the UTF-8 encoding.

A typical usage pattern of this API consists of first calling `MultiByteToWideChar` to get the size of the result string. Then, some string buffer is allocated according to that size value. This is typically done using the `std::wstring::resize` method in case the destination is a UTF-16 string. (For further details, you may want to read my July 2015 article, “Using STL Strings at Win32 API Boundaries” at msdn.com/magazine/mt238407.) Finally, the `MultiByteToWideChar` function is invoked a second time to do the actual encoding conversion, using the destination string buffer previously allocated. Note that the *same* usage pattern applies to the symmetric `WideCharToMultiByte` API.

Let's implement this pattern in C++ code, inside the body of the custom `Utf8ToUtf16` conversion function. Start handling the special case of an empty input string, where just an empty output `wstring` is returned:

```
#include <Windows.h> // For Win32 APIs
#include <string> // For std::string and std::wstring

std::wstring Utf8ToUtf16(const std::string& utf8)
{
    std::wstring utf16; // Result

    if (utf8.empty())
    {
        return utf16;
    }
```

Conversion Flags `MultiByteToWideChar` can be called for the first time to get the size of the destination UTF-16 string. This Win32 function has a relatively complex interface, and its behavior is defined according to some flags. Because this API will be called twice in the `Utf8ToUtf16` conversion function's body, it's good practice for code readability and maintainability to define a named constant that can be used in both calls:

```
// Safely fails if an invalid UTF-8 character
// is encountered in the input string
constexpr DWORD kFlags = MB_ERR_INVALID_CHARS;
```

It's also good practice from a security perspective to make the conversion process fail if an invalid UTF-8 sequence is found in the input string. The use of the `MB_ERR_INVALID_CHARS` flag is also encouraged in Michael Howard and David LeBlanc's book, "Writing Secure Code, Second Edition" (Microsoft Press, 2003).

If your project uses an older Visual C++ compiler version that doesn't support the `constexpr` keyword, you can substitute `static const` in that context.

String Lengths and Safe Conversions from `size_t` to `int` `MultiByteToWideChar` expects the input string length parameter expressed using the `int` type, while the `length` method of the STL string classes returns a value of type equivalent to `size_t`. In 64-bit builds, the Visual C++ compiler emits a warning signaling a potential loss of data for the conversion from `size_t` (whose size is 8 bytes) to `int` (which is 4 bytes in size). But even in 32-bit builds, where both `size_t` and `int` are defined as 32-bit integers by the Visual C++ compiler, there's an unsigned/signed mismatch: `size_t` is unsigned, while `int` is signed. This isn't a problem for strings of reasonable length, but for gigantic strings of length greater than ($2^{31}-1$)—that is, more than 2 billion bytes in size—the conversion from an unsigned integer (`size_t`) to a signed integer (`int`) can generate a negative number, and negative lengths don't make sense.

So, instead of just invoking `utf8.length` to get the size of the source UTF-8 input string, and passing that to the `MultiByteToWideChar` API, it's better to check the actual `size_t`-value of the length, making sure it can be safely and meaningfully converted to an `int`, and only then pass it to the `MultiByteToWideChar` API.

The following code can be used to make sure that `size_t`-length doesn't exceed the maximum value for a variable of type `int`, throwing an exception if it does:

```
if (utf8.length() > static_cast<size_t>(std::numeric_limits<int>::max()))
{
    throw std::overflow_error(
        "Input string too long: size_t-length doesn't fit into int.");
}
```

Note the use of the `std::numeric_limits` class template (from the `<limits>` C++ standard header) to query the largest possible value for the type `int`. However, this code might actually not compile. How's that? The problem is in the definition of the `min` and `max` macros in the Windows Platform SDK headers. In particular, the Windows-specific definition of the `max` preprocessor macro conflicts with the `std::numeric_limits<int>::max` member function call. There are some ways to prevent that.

A possible solution is to `#define NOMINMAX` before including `<Windows.h>`. This will prevent the definition of the `min` and `max` Windows-specific preprocessor macros. However, preventing the definition of these macros may actually cause problems with other

Windows headers, such as `<gdiplus.h>`, which do require the definitions of these Windows-specific macros.

So, another option is to use an extra pair of parentheses around the `std::numeric_limits::max` member function call, to prevent against the aforementioned macro expansion:

```
if (utf8.length() > static_cast<size_t>((std::numeric_limits<int>::max)
()))
{
    throw std::overflow_error(
        "Input string too long: size_t-length doesn't fit into int.");
}
```

Moreover, as an alternative, the `INT_MAX` constant could be used instead of the C++ `std::numeric_limits` class template.

Whatever approach is used, once the size check is done and the length value is found to be suitable for a variable of type `int`, the cast from `size_t` to `int` can be safely performed using `static_cast`:

```
// Safely convert from size_t (STL string's length)
// to int (for Win32 APIs)
const int utf8Length = static_cast<int>(utf8.length());
```

Note that the UTF-8 string's length is measured in 8-bit char units; that is, in bytes.

First API Call: Getting the Destination String's Length Now `MultiByteToWideChar` can be called for the first time, to get the destination UTF-16 string's length:

```
const int utf16Length = ::MultiByteToWideChar(
    CP_UTF8,          // Source string is in UTF-8
    kFlags,           // Conversion flags
    utf8.data(),       // Source UTF-8 string pointer
    utf8Length,        // Length of the source UTF-8 string, in chars
    nullptr,          // Unused - no conversion done in this step
    0,                // Request size of destination buffer, in wchar_ts
);
```

Note how the function is invoked passing zero as the last argument. This instructs the `MultiByteToWideChar` API to just return the required size for the destination string; no conversion is done in this step. Note also that the size of the destination string is expressed in `wchar_ts` (not in 8-bit chars), which makes sense, because the destination string is a UTF-16-encoded Unicode string, made by sequences of 16-bit `wchar_ts`.

To get read-only access to the input UTF-8 `std::string`'s content, the `std::string::data` method is called. Because the UTF-8 string's length is explicitly passed as an input parameter, this code will work also for instances of `std::string` having embedded NULs inside.

Note also the use of the `CP_UTF8` constant to specify that the input string is encoded in UTF-8.

Handling the Error Case If the preceding function call fails, for example, in presence of invalid UTF-8 sequences in the input string, the `MultiByteToWideChar` API returns zero. In this case, the `GetLastError` Win32 function can be invoked to get further details about the cause of the failure. A typical error code returned in case of invalid UTF-8 characters is `ERROR_NO_UNICODE_TRANSLATION`.

In case of failure, it's time to throw an exception. This can be an instance of the previously custom-designed `Utf8ConversionException` class:

```
if (utf16Length == 0)
{
    // Conversion error: capture error code and throw
    const DWORD error = ::GetLastError();
    throw Utf8ConversionException(
        "Cannot get result string length when converting " \
        "from UTF-8 to UTF-16 (MultiByteToWideChar failed).",
        error);
}
```


Manipulating Files?

View, annotate, compare, convert, assemble, sign and share over **50 types of documents on the web.**

GroupDocs.Viewer
GroupDocs.Annotation
GroupDocs.Conversion
GroupDocs.Comparison
GroupDocs.Signature
GroupDocs.Assembly
GroupDocs.Metadata
GroupDocs.Search



[Get started now](#)



.NET Libraries



Java Libraries



Cloud APIs



Cloud Apps

Contact Us:

US: +1 903 306 1676

EU: +44 141 628 8900

AU: +61 2 8006 6987

sales@groupdocs.com

Visit us at www.groupdocs.com

Allocating Memory for the Destination String If the Win32 function call succeeds, the required destination string length is stored in the `utf16Length` local variable, so the destination memory for the output UTF-16 string can be allocated. For UTF-16 strings stored in instances of the `std::wstring` class, a simple call to the `resize` method would be just fine:

```
utf16.resize(utf16Length);
```

Note that because the length of the input UTF-8 string was explicitly passed to `MultiByteToWideChar` (instead of just passing `-1` and asking the API to scan the whole input string until a NUL-terminator is found), the Win32 API won't add an additional NUL-terminator to the resulting string: The API will just process the exact number of chars in the input string specified by the explicitly passed length value. Therefore, there's no need to invoke `std::wstring::resize` with a `"utf16Length + 1"` value: Because no additional NUL-terminator will be scribbled in by the Win32 API, you don't have to make room for it inside the destination `std::wstring` (more details on that can be found in my July 2015 article).

Second API Call: Doing the Actual Conversion Now that the UTF-16 `wstring` instance has enough space to host the resulting UTF-16 encoded text, it's finally time to call `MultiByteToWideChar` for the second time, to get the actual converted bits in the destination string:

```
// Convert from UTF-8 to UTF-16
int result = ::MultiByteToWideChar(
    CP_UTF8,          // Source string is in UTF-8
    kFlags,           // Conversion flags
    utf8.data(),      // Source UTF-8 string pointer
    utf8Length,       // Length of source UTF-8 string, in chars
    &utf16[0],         // Pointer to destination buffer
    utf16Length       // Size of destination buffer, in wchar_ts
);
```

Note the use of the `"&utf16[0]"` syntax to gain write access to the `std::wstring`'s internal memory buffer (this, too, was already discussed in my July 2015 article).

If the first call to `MultiByteToWideChar` succeeds, it's unlikely this second call will fail. Still, checking the API return value is certainly a good, safe coding practice:

```
if (result == 0)
{
    // Conversion error: capture error code and throw
    const DWORD error = ::GetLastError();
    throw Utf8ConversionException(
        "Cannot convert from UTF-8 to UTF-16 "\
        "(MultiByteToWideChar failed).",
        error);
}
```

Else, in case of success, the resulting UTF-16 string can finally be returned to the caller:

```
return utf16;

} // End of Utf8ToUtf16
```

Usage Sample So, if you have a UTF-8-encoded Unicode string (for example, coming from some C++ cross-platform code) and you want to pass it to a Win32 Unicode-enabled API, this custom conversion function can be invoked like so:

```
std::string utf8Text = /* ...some UTF-8 Unicode text ... */;

// Convert from UTF-8 to UTF-16 at the Win32 API boundary
::SetWindowText(myWindow, Utf8ToUtf16(utf8Text).c_str());

// Note: In Unicode builds (Visual Studio default) SetWindowText
// is expanded to SetWindowTextW
```

The `Utf8ToUtf16` function returns a `wstring` instance containing the UTF-16-encoded string, and the `c_str` method is invoked on this instance to get a C-style raw pointer to a NUL-terminated string, to be passed to Win32 Unicode-enabled APIs.

Very similar code can be written for the reverse conversion from UTF-16 to UTF-8, this time calling the `WideCharToMultiByte` API. As I noted earlier, Unicode conversions between UTF-8 and UTF-16 are lossless—no characters will be lost during the conversion process.

Unicode Encoding Conversion Library

Sample compilable C++ code is included in the downloadable archive associated with this article. This is reusable code, compiling cleanly at the Visual C++ warning level 4 (/W4) in both 32-bit and 64-bit builds. It's implemented as a header-only C++ library. Basically, this Unicode encoding conversion module consists of two header files: `utf8except.h` and `utf8conv.h`. The former contains the definition of a C++ exception class used to signal error conditions during the Unicode encoding conversions. The latter implements the actual Unicode encoding conversion functions.

Note that `utf8except.h` contains only cross-platform C++ code, making it possible to catch the UTF-8 encoding conversion exception anywhere in your C++ projects, including portions of code that aren't Windows-specific, and instead use cross-platform C++ by design. In contrast, `utf8conv.h` contains C++ code that's Windows-specific, because it directly interacts with the Win32 API boundary.

To reuse this code in your projects, just `#include` the aforementioned header files. The downloadable archive contains an additional source file implementing some test cases, as well.

Wrapping Up

Unicode is the de facto standard for representing international text in modern software. Unicode text can be encoded in various formats: The two most important ones are UTF-8 and UTF-16. In C++ Windows code there's often a need to convert between UTF-8 and UTF-16, because Unicode-enabled Win32 APIs use UTF-16 as their native Unicode encoding. UTF-8 text can be conveniently stored in instances of the STL `std::string` class, while `std::wstring` is well-suited to store UTF-16-encoded text in Windows C++ code targeting the Visual C++ compiler.

The Win32 APIs `MultiByteToWideChar` and `WideCharToMultiByte` can be used to perform conversions between Unicode text represented using the UTF-8 and UTF-16 encodings. I showed a detailed description of the usage pattern of the `MultiByteToWideChar` API, wrapping it in a reusable modern C++ helper function to perform conversions from UTF-8 to UTF-16. The reverse conversion follows a very similar pattern, and reusable C++ code implementing it is available in this article's download. ■

GIOVANNI DICANIO is a computer programmer specializing in C++ and Windows, a Pluralsight author and a Visual C++ MVP. Besides programming and course authoring, he enjoys helping others on forums and communities devoted to C++, and can be contacted at giovanni.dicanio@gmail.com. He also writes a blog at blogs.msmvps.com/gdicanio.

THANKS to the following technical experts for reviewing this article:
David Cravey and Marc Gregoire

JOIN US on the CAMPAIGN TRAIL in 2016!



SEPTEMBER 26 - 29

HYATT ORANGE COUNTY, CA –
A DISNEYLAND® GOOD
NEIGHBOR HOTEL

vslive.com/anaheim

See pages 52–53 for more info



OCTOBER 3 - 6

RENAISSANCE, WASHINGTON, D.C.

vslive.com/dc

See pages 54–55 for more info



PART OF LIVE! 360



DECEMBER 5 - 9

ROYAL PACIFIC RESORT
AT UNIVERSAL ORLANDO
ORLANDO, FL

vslive.com/orlando

See pages 70–73 for more info



Look for the special agenda
pull-out section on page 33



6 GREAT CONFERENCES. 1 GREAT PRICE.

Visual Studio **LIVE!**

SQL Server **LIVE!**

TECHMENTOR

Office &
SharePoint **LIVE!**

ModernApps **LIVE!**

APPDEV
TRENDS **NEW!**

**Orlando
2016**

LIVE360EVENTS.COM

CONNECT WITH VISUAL STUDIO LIVE!



twitter.com/vslive – @VSLive



facebook.com – Search “VSLive”



linkedin.com – Join the
“Visual Studio Live” group!

VSLIVE.COM

CAMPAIGN FOR CODE 2016 ★ VISUAL STUDIO LIVE!

CODE

Anaheim

FOR A BETTER TOMORROW



SEPTEMBER 26-29, 2016

**HYATT REGENCY
DISNEYLAND®**

GOOD NEIGHBOR HOTEL

VISUAL STUDIO LIVE! (VSLive!™) is blazing new trails on the Campaign for Code: For the first time ever, we are headed to Anaheim, CA, to bring our unique brand of unbiased developer training to Southern California. With our host hotel situated just down the street from Disneyland®, developers, software architects, engineers, designers and more can code by day, and visit the Magic Kingdom by night. From Sept. 26-29, explore how the code you write today will create a better tomorrow—not only for your company, but your career, as well!

**California code with us:
register to join us today!**

**SESSIONS ARE FILLING UP QUICKLY—
REGISTER TODAY!**

Scan the QR code to register or
for more event details.



USE PROMO CODE VSLANS

SUPPORTED BY



Visual Studio
MAGAZINE



PRODUCED BY



ANAHEIM AGENDA AT-A-GLANCE

ALM / DevOps	ASP.NET	Cloud Computing	Database and Analytics	JavaScript / HTML5 Client	Mobile Client	Software Practices	Visual Studio / .NET Framework	Windows Client
Visual Studio Live! Pre-Conference Workshops: Monday, September 26, 2016 (Separate entry fee required)								
START TIME	END TIME	Pre-Conference Workshop Registration - Coffee and Morning Pastries						
7:30 AM	9:00 AM							
9:00 AM	6:00 PM	M01 Workshop: Building for the Internet of Things: Hardware, Sensors & the Cloud - Nick Landry	M02 Workshop: Developer Dive into SQL Server 2016 - Leonard Lobel	M03 Workshop: Distributed Cross-Platform Application Architecture - Rockford Lhotka & Jason Bock			M04 Workshop: Creating Awesome 2D & 3D Games and Experiences with Unity - Adam Tuliper	
6:45 PM	9:00 PM	Dine-A-Round						
Visual Studio Live! Day 1: Tuesday, September 27, 2016								
START TIME	END TIME	Registration - Coffee and Morning Pastries						
7:00 AM	8:00 AM							
8:00 AM	9:00 AM	Keynote: To Be Announced						
9:15 AM	10:30 AM	T01 Angular 2 101 - Deborah Kurata	T02 Developing Universal Windows Platform (UWP) Applications - Scott Golightly	T03 Developer Productivity in Visual Studio 2015 - Robert Green			☀ T04 This session is sequestered, details will be released soon	
10:45 AM	12:00 PM	T05 ASP.NET MVC6—What You Need to Know - Philip Japikse	T06 Building Connected and Disconnected Mobile Applications - James Montemagno	T07 Professional Scrum Development Using Visual Studio 2015 - Richard Hundhausen			☀ T08 This session is sequestered, details will be released soon	
12:00 PM	1:30 PM	Lunch - Visit Exhibitors						
1:30 PM	2:45 PM	T09 Angular 2 Forms and Validation - Deborah Kurata	T10 New SQL Server 2016 Security Features for Developers - Leonard Lobel	T11 Use Visual Studio to Scale Agile in Your Enterprise - Richard Hundhausen			T12 Windows for Makers: Raspberry Pi, Arduino & IoT - Nick Landry	
3:00 PM	4:15 PM	T13 Introduction to ASP.NET Core 1.0 - Scott Golightly	T14 Cross-platform Mobile Development for the C# Developer - James Montemagno	☀ T15 This session is sequestered, details will be released soon			T16 Cloud Enable .NET Client LOB Applications - Robert Green	
4:15 PM	5:30 PM	Welcome Reception						
Visual Studio Live! Day 2: Wednesday, September 28, 2016								
START TIME	END TIME	Registration - Coffee and Morning Pastries						
7:00 AM	8:00 AM							
8:00 AM	9:15 AM	W01 JavaScript Patterns for the C# Developer - Ben Hoelting	W02 Implementing the Mvvm Pattern in Your Xamarin Apps - Kevin Ford	W03 VSTS and Azure: Cloud DevOps 101 - Mike Benkovich			W04 Introduction to Next Generation of Azure PaaS—Service Fabric and Containers - Vishwas Lele	
9:30 AM	10:45 AM	W05 Busy Developer's Guide to TypeScript - Ted Neward	☀ W06 This session is sequestered, details will be released soon	W07 User Story Mapping - Philip Japikse			W08 Cloud Oriented Programming - Vishwas Lele	
11:00 AM	12:00 PM	GENERAL SESSION: FAST, EASY, AND FUN—MOBILE DEVELOPMENT FOR EVERYONE WITH XAMARIN - James Montemagno, Developer Evangelist, Xamarin Team, Microsoft						
12:00 PM	1:30 PM	Birds-of-a-Feather Lunch - Visit Exhibitors						
1:30 PM	2:45 PM	W09 Getting Started with ASP.NET Core 1 Web API and Entity Framework Core 1 - Philip Japikse	W10 Using Visual Studio Tools for Apache Cordova to Create MultiPlatform Applications - Kevin Ford	W11 Exploring C# 7 New Features - Adam Tuliper			W12 Lock the Doors, Secure the Valuables, and Set the Alarm - Eric D. Boyd	
3:00 PM	4:15 PM	W13 Angular 2 and ASP.NET Core—A Perfect Match! - Dan Wahlin	W14 Busy .NET Developer's Guide to Swift - Ted Neward	W15 I'm Emotional - Using Microsoft Cognitive Services to Understand the World Around You - Adam Tuliper			W16 App Services: The New PaaS - Mike Benkovich	
4:30 PM	5:45 PM	W17 Increase Website Performance and Search with Lucene.Net Indexing - Ben Hoelting	W18 Reusing Business Logic in Cross-Platform .NET - Rockford Lhotka	W19 Database Development with SQL Server Data Tools - Leonard Lobel			W20 Breaking Down Walls with Modern Identity - Eric D. Boyd	
7:00 PM	9:00 PM	Evening Event						
Visual Studio Live! Day 3: Thursday, September 29, 2016								
START TIME	END TIME	Registration - Coffee and Morning Pastries						
7:00 AM	8:00 AM							
8:00 AM	9:15 AM	TH01 Building Data-Centric Apps with Angular 2 and Breeze - Brian Noyes	TH02 Building Business Apps on the Universal Windows Platform - Billy Hollis	TH03 Using Your SQL Skills to Become a Big Data Developer - Omid Afnan			TH04 Docker: Creating the Ultimate Development Environment - Dan Wahlin	
9:30 AM	10:45 AM	TH05 Getting Started with Aurelia - Brian Noyes	TH06 Porting Desktop and Tablet Apps in Windows 10 to the HoloLens - Billy Hollis	TH07 Data Analytics with Azure Data Lake: U-SQL In Depth - Omid Afnan			H08 DevOps for Developers - Brian Randell	
11:00 AM	12:15 PM	TH09 Pretty, Yet Powerful. How Data Visualization Transforms the Way We Comprehend Information - Walt Ritscher	TH10 Dependencies Demystified - Jason Bock	TH11 Predicting the Future Using Azure Machine Learning - Eric D. Boyd			TH12 PowerShell for Developers - Brian Randell	
12:15 PM	1:30 PM	Lunch						
1:30 PM	2:45 PM	TH13 Building Rich UI with ASP.NET MVC and Bootstrap - Walt Ritscher	TH14 What's the Problem?!?: How to Resolve Conflict - Robert Bogue	TH15 Exploratory Data Analysis with R Tools for Visual Studio - Matthew Renze			TH16 Developing Awesome 3D Apps with Unity and C# - Adam Tuliper	
3:00 PM	4:15 PM	TH17 Assembling the Web - A Tour of WebAssembly - Jason Bock	TH18 Hack Proof: Software Design for a Hostile Internet - Robert Bogue	TH19 Data Visualization with R Tools for Visual Studio - Matthew Renze			TH20 Technical Debt - Fight It with Science and Rigor - Brian Randell	

Speakers and sessions subject to change



DETAILS COMING SOON! These sessions have been sequestered by our conference chairs. Be sure to check vslive.com/anaheim for session updates!

CONNECT WITH VISUAL STUDIO LIVE!



twitter.com/vslive - @VSLive



facebook.com - Search "VSLive"



linkedin.com - Join the "Visual Studio Live" group!

VSLIVE.COM/ANAHEIM

VOTE "YES" FOR BETTER

CAMPAIGN FOR CODE 2016 ★ VISUAL STUDIO LIVE!

CODE Washington, D.C.

OCT. 3-6, 2016

**RENAISSANCE
WASHINGTON, D.C.**



VISUAL STUDIO LIVE! (VSLive!™) is on a Campaign for Code in 2016, in support of developer education. It's only fitting that we return with our unique brand of practical, unbiased, Developer training to the nation's capital this year. From Oct. 3-6, we're offering four days of sessions, workshops and networking events to developers, software architects, engineers and designers—all designed to help you vote "yes" for better code and write winning applications across all platforms.

**Do your developer duty:
register to join us today!**

**REGISTER BY SEPTEMBER 7
& SAVE \$200!**

Scan the QR code to register or
for more event details.



USE PROMO CODE VSLDC5

SUPPORTED BY



Visual Studio
MAGAZINE



PRODUCED BY



WASHINGTON, D.C. AGENDA AT-A-GLANCE

ALM / DevOps	Cloud Computing	Database and Analytics	Mobile Client	Software Practices	Visual Studio / .NET Framework	Web Client	Web Server	Windows Client
Visual Studio Live! Pre-Conference Workshops: Monday, October 3, 2016 (Separate entry fee required)								
START TIME	END TIME	Pre-Conference Workshop Registration • Coffee and Morning Pastries						
7:30 AM	9:00 AM							
9:00 AM	6:00 PM	M01 Workshop: DevOps in a Day - Brian Randell		M02 Workshop: SQL Server for Developers - Andrew Brust and Leonard Lobel		M03 Workshop: Distributed Cross-Platform Application Architecture - Rockford Lhotka & Jason Bock		
6:45 PM	9:00 PM	Dine-A-Round						
Visual Studio Live! Day 1: Tuesday, October 4, 2016								
START TIME	END TIME	Registration - Coffee and Morning Pastries						
7:00 AM	8:00 AM							
8:00 AM	9:00 AM	Keynote: To Be Announced						
9:15 AM	10:30 AM	T01 AngularJS2 in 75 Minutes - Sahil Malik	T02 This session is sequestered, details will be released soon		T03 Developer Productivity in Visual Studio 2015 - Robert Green		T04 Cloud Oriented Programming - Vishwas Lele	
10:45 AM	12:00 PM	T05 ASP.NET MVC6—What You Need to Know - Philip Japikse	T06 What's New in SQL Server 2016 - Leonard Lobel		T07 HoloLens - Brian Randell		T08 This session is sequestered, details will be released soon	
12:00 PM	1:30 PM	Lunch - Visit Exhibitors						
1:30 PM	2:45 PM	T09 AngularJS2 with Web, Mobile (Cordova), and Desktop (electron) - Sahil Malik	T10 Database Development with SQL Server Data Tools - Leonard Lobel		T11 This session is sequestered, details will be released soon		T12 Windows for Makers: Raspberry Pi, Arduino & IoT - Nick Landry	
3:00 PM	4:15 PM	T13 Getting Started with ASP.NET Core 1 Web API and Entity Framework Core 1 - Philip Japikse	T14 Cross-platform Mobile Development for the C# Developer - James Montemagno		T15 Exploring C# 7 New Features - Adam Tuliper		T16 Cloud Enable .NET Client LOB Applications - Robert Green	
4:15 PM	5:30 PM	Welcome Reception						
Visual Studio Live! Day 2: Wednesday, October 5, 2016								
START TIME	END TIME	Registration - Coffee and Morning Pastries						
7:00 AM	8:00 AM							
8:00 AM	9:15 AM	W01 JavaScript Patterns for the C# Developer - Ben Hoelting	W02 Implementing the Mvvm Pattern in Your Xamarin Apps - Kevin Ford		W03 Developing Awesome 3D Apps with Unity and C# - Adam Tuliper		W04 Introduction to Next Generation of Azure PaaS—Service Fabric and Containers - Vishwas Lele	
9:30 AM	10:45 AM	W05 Busy Developer's Guide to TypeScript - Ted Neward	W06 Building Connected and Disconnected Mobile Applications - James Montemagno		W07 Unleash the New .NET: Open Source and Running on OS X, Linux & Windows - Nick Landry		W08 This session is sequestered, details will be released soon	
11:00 AM	12:00 PM	GENERAL SESSION: More Personal Computing through Emerging Experiences - Tim Huckaby, Founder / Chairman - Interknowlogy & Actus Interactive Software						
12:00 PM	1:30 PM	Birds-of-a-Feather Lunch - Visit Exhibitors						
1:30 PM	2:45 PM	W09 WCF & Web API: Can We All Just Get Along?!? - Miguel Castro	W10 Using Visual Studio Tools for Apache Cordova to Create MultiPlatform Applications - Kevin Ford		W11 Software Engineering in an Agile Environment - David Corbin		W12 Lock the Doors, Secure the Valuables, and Set the Alarm - Eric D. Boyd	
3:00 PM	4:15 PM	W13 Richer MVC Sites with Knockout JS - Miguel Castro	W14 Busy .NET Developer's Guide to Swift - Ted Neward		W15 Technical Debt—Fight It with Science and Rigor - Brian Randell		W16 Breaking Down Walls with Modern Identity - Eric D. Boyd	
4:30 PM	5:45 PM	W17 Increase Website Performance and Search with Lucene.Net Indexing - Ben Hoelting	W18 Reusing Business Logic in Cross-Platform .NET - Rockford Lhotka		W19 Implementing Data Protection and Security in SQL Server 2016 - Steve Jones		W20 Build Real-Time Websites and Apps with SignalR and Azure - Rachel Appel	
6:45 PM	10:30 PM	Visual Studio Live! Monuments by Moonlight Tour						
Visual Studio Live! Day 3: Thursday, October 6, 2016								
START TIME	END TIME	Registration - Coffee and Morning Pastries						
7:00 AM	8:00 AM							
8:00 AM	9:15 AM	TH01 Building Data-Centric Apps with Angular 2 and Breeze - Brian Noyes	TH02 Moving from WPF to Windows 10 and Universal Apps - Billy Hollis		TH03 Test Drive Automated GUI Testing with WebDriver - Rachel Appel		TH04 Big Data and Hadoop with Azure HDInsight - Andrew Brust	
9:30 AM	10:45 AM	TH05 Getting Started with Aurelia - Brian Noyes	TH06 Building Business Apps on the Universal Windows Platform - Billy Hollis		TH07 Exposing an Extensibility API for your Applications - Miguel Castro		TH08 Power BI 2.0: Analytics in the Cloud and in Excel - Andrew Brust	
11:00 AM	12:15 PM	TH09 Dependencies Demystified - Jason Bock	TH10 Pretty, Yet Powerful. How Data Visualization Transforms the Way We Comprehend Information - Walt Ritscher		TH11 Bringing DevOps to the Database - Steve Jones		TH12 Predicting the Future Using Azure Machine Learning - Eric D. Boyd	
12:15 PM	1:30 PM	Lunch						
1:30 PM	2:45 PM	TH13 Building Rich UI with ASP.NET MVC and Bootstrap - Walt Ritscher	TH14 What's the Problem?!?: How to Resolve Conflict - Robert Bogue		TH15 Build Distributed Business Apps Using CSLA .NET - Rockford Lhotka		TH16 Exploratory Data Analysis with R Tools for Visual Studio - Matthew Renze	
3:00 PM	4:15 PM	TH17 Assembling the Web—A Tour of WebAssembly - Jason Bock	TH18 Hack Proof: Software Design for a Hostile Internet - Robert Bogue		TH19 Bootstrapping Automated Testing for Existing Software Systems - David Corbin		TH20 Data Visualization with R Tools for Visual Studio - Matthew Renze	

Speakers and sessions subject to change

★ DETAILS COMING SOON! These sessions have been sequestered by our conference chairs. Be sure to check vslive.com/dc for session updates!

CONNECT WITH VISUAL STUDIO LIVE!



twitter.com/vslive – @VSLive



[facebook.com](https://www.facebook.com/vslive) – Search “VSLive”



[linkedin.com](https://www.linkedin.com/groups?gid=11111111) – Join the “Visual Studio Live” group!

VSLIVE.COM/DC



The Secretary Problem

Suppose you want to hire a secretary. You have a pool of 100 applicants and you interview one applicant per day. After each interview, you must immediately decide to hire the current applicant or not. If you don't hire an applicant, you can't call her back. You don't have time to interview all 100 candidates, so what algorithm can you use to maximize your chance of selecting the best applicant?

What I've just described is one example of the Secretary Problem. Solutions to the Secretary Problem have many important uses. For example, in machine learning, it's often necessary to come up with some approach for stopping training in a way to optimize the probability of selecting the best prediction model.

In machine learning, it's often necessary to come up with some approach for stopping training in a way to optimize the probability of selecting the best prediction model.

The Secretary Problem is part of a larger class of problems sometimes called best choice problems, and also part of what are called optimal stopping problems. To the best of my knowledge, a description of the Secretary Problem (in a slightly different form) was first published in *Scientific American* magazine in 1960. There are dozens of interesting variations of the problem.

In this article I'll show you how to tackle the Secretary Problem using what's called the $1/e$ stopping rule. A good way to see where this article is headed is to examine the demo program in **Figure 1**. I coded the demo using C#, but you shouldn't have any trouble refactoring the code to another language if you wish.

This article assumes you have at least intermediate programming ability, but doesn't assume you know anything about the Secretary Problem. The complete source code for the demo program is presented in this article, and you can also get the source from the accompanying code download.

Code download available at msdn.com/magazine/0916magcode.

The $1/e$ Stopping Rule

It can be mathematically proved that, subject to certain conditions, you can theoretically maximize the probability of selecting the best applicant in the Secretary Problem by using what's called the $1/e$ rule or algorithm. In the context of the Secretary Problem, the term Candidate means the best applicant seen at a given point in time. I capitalize the word Candidate to emphasize that the word has a different meaning than in normal English, where a candidate and an applicant are essentially synonymous.

In the following explanation, N represents the total number of applicants, and e represents Euler's constant, which is approximately 2.71828. In words, the $1/e$ rule is, "Skip over the first N/e applicants, but track the best Candidate. Then hire the first Candidate who appears. If no new Candidate appears after the first N/e applicants have been skipped, then fail and hire nobody."

A concrete example should make the algorithm clear. The $1/e$ stopping algorithm is illustrated graphically in **Figure 2**. Suppose you have 10 applicants. Each applicant has (unknown to you until after you interview them) a rating between 0.0 and 9.0, where higher values are better. The applicants' ratings are:

(5.0,	2.0,	7.0,	1.0,	4.0,	0.0,	8.0,	3.0,	9.0,	6.0)
0	1	2	3	4	5	6	7	8	9

Applicant 0 has a rating of 5.0 and will be interviewed first; applicant 1 has a rating of 2.0 and will be interviewed second; and so on. The best applicant is person 8 with a rating of 9.0.

The number of applicants to skip is $N/e = 10 / 2.71828 = 3.6788$, which is 3 if truncated, and 4 if rounded. As it turns out, as long as N is not very small it makes very little difference whether you truncate or round. Suppose you truncate to 3.

Solutions to the Secretary Problem have many important uses.

You interview applicant 0 and find she has a rating of 5.0, so she becomes the Candidate because she has the best rating seen (so far, the only rating seen). Next, you interview applicant 1 and find they have a rating of 2.0 so they don't become the Candidate because their rating isn't better than 5.0. You interview applicant 2 and find a rating of 7.0 and they become the new Candidate. At this point you've interviewed the first $N/e = 3$ applicants so you're now ready to hire the first new Candidate who appears.

msdn

magazine

Where you need us most.



Visual Studio **LIVE!**
EXPERT SOLUTIONS FOR .NET DEVELOPERS

LIVE!
360
TECH EVENTS WITH PERSPECTIVE

MSDN.microsoft.com

You interview applicant 3 and find a rating of 1.0 so they don't become the Candidate. You interview applicant 5 and find a rating of 0.0 (Ouch! I've worked with this person) so they're not the Candidate, either. You interview applicant 6 and find a rating of 8.0. This is the highest rating seen so they become the Candidate, and because you're past the first N/e applicants, you immediately hire applicant 6.

Notice that the $1/e$ algorithm did not find the best applicant in this example, but did find the second best applicant. If you use the $1/e$ algorithm for the Secretary Problem, the probability that you'll select the best applicant from N applicants is approximately $1/e = 1/2.71828 = 0.3679$.

The Demo Program

To create the demo program, I launched Visual Studio and made a new C# console application program named SecretaryProblem. The demo program has no significant .NET dependencies, so any version of Visual Studio will work. After the template code loaded, in the Solution Explorer window, I right-clicked on file Program.cs and renamed it to the more-descriptive SecretaryProblemProgram.cs and Visual Studio automatically renamed class Program for me. At the top of the Editor window, I removed all using statements that referenced unnecessary namespaces, leaving just the one reference to the top-level System namespace.

The basic version of the Secretary Problem assumes that all applicant ratings are different, so there can be no ties.

The demo program has two parts. The first few lines in the Main method illustrate the $1/e$ algorithm when applied to a specific Secretary Problem example with 10 applicants. The second part of the demo illustrates a simulation where the $1/e$ algorithm is run 10,000 times with 100 applicants.

The key lines of calling code for the first part of the demo are:

```
double[] ratings =
    new double[] { 5.0, 2.0, 7.0, 1.0, 4.0, 0.0, 8.0, 3.0, 9.0, 6.0 };

Console.WriteLine("Applicant ratings are: \n");
for (int i = 0; i < ratings.Length; ++i)
    Console.WriteLine(ratings[i].ToString("F1") + " ");
Console.WriteLine("\n");
```

```
int selected = Select(ratings, true); // Verbose
```

A problem is set up by defining applicants as an array of type double where the index represents an applicant ID, and the cell value represents an

applicant's rating. The basic version of the Secretary Problem assumes that all applicant ratings are different, so there can be no ties.

The program-defined Select method accepts a rating array and uses the $1/e$ algorithm to find and return the index of the selected applicant. The Select method has a lot of WriteLine statements that print progress, as shown in Figure 1.

The key lines of calling code (with minor editing for readability) for the second part of the demo are:

```
int numHiringSessions = 10000;
int numApplicants = 100;

double pBest = Simulation(numApplicants, numHiringSessions);
Console.WriteLine("Simulation prob of best applicant = " +
    pBest.ToString("F4"));

double pTheory = 1 / Math.E;
Console.WriteLine("Theoretical prob of best applicant = " +
    pTheory.ToString("F4"));
```

The program-defined Simulation method accepts a fixed number of applicants. An alternative design is to allow the simulation

```
file:///C:/SecretaryProblem/bin/Debug/SecretaryProblem.EXE
Begin Secretary Problem <best choice> demo
Applicant ratings are:
5.0 2.0 7.0 1.0 4.0 0.0 8.0 3.0 9.0 6.0
numApplicants = 10
numSkip = 3
=====
Applicant = 0
Rating = 5.0
Ready to hire = False
Applicant is not a candidate
=====
Applicant = 1
Rating = 2.0
Ready to hire = False
Applicant is not a candidate
=====
Applicant = 2
Rating = 7.0
Ready to hire = False
Applicant is the new candidate
Not ready to hire. Moving to next applicant
=====
Applicant = 3
Rating = 1.0
Ready to hire = True
Applicant is not a candidate
=====
Applicant = 4
Rating = 4.0
Ready to hire = True
Applicant is not a candidate
=====
Applicant = 5
Rating = 0.0
Ready to hire = True
Applicant is not a candidate
=====
Applicant = 6
Rating = 8.0
Ready to hire = True
Applicant is the new candidate
Hiring this applicant

Begin simulation
Number of simulated applicants = 100
Number of simulated hiring sessions = 10000
Number of times best applicant selected = 3726
Simulation probability of selecting best applicant = 0.3726
Theoretical probability of selecting best applicant = 0.3679

End Secretary Problem demo
```

Figure 1 Secretary Problem Demo Program Run

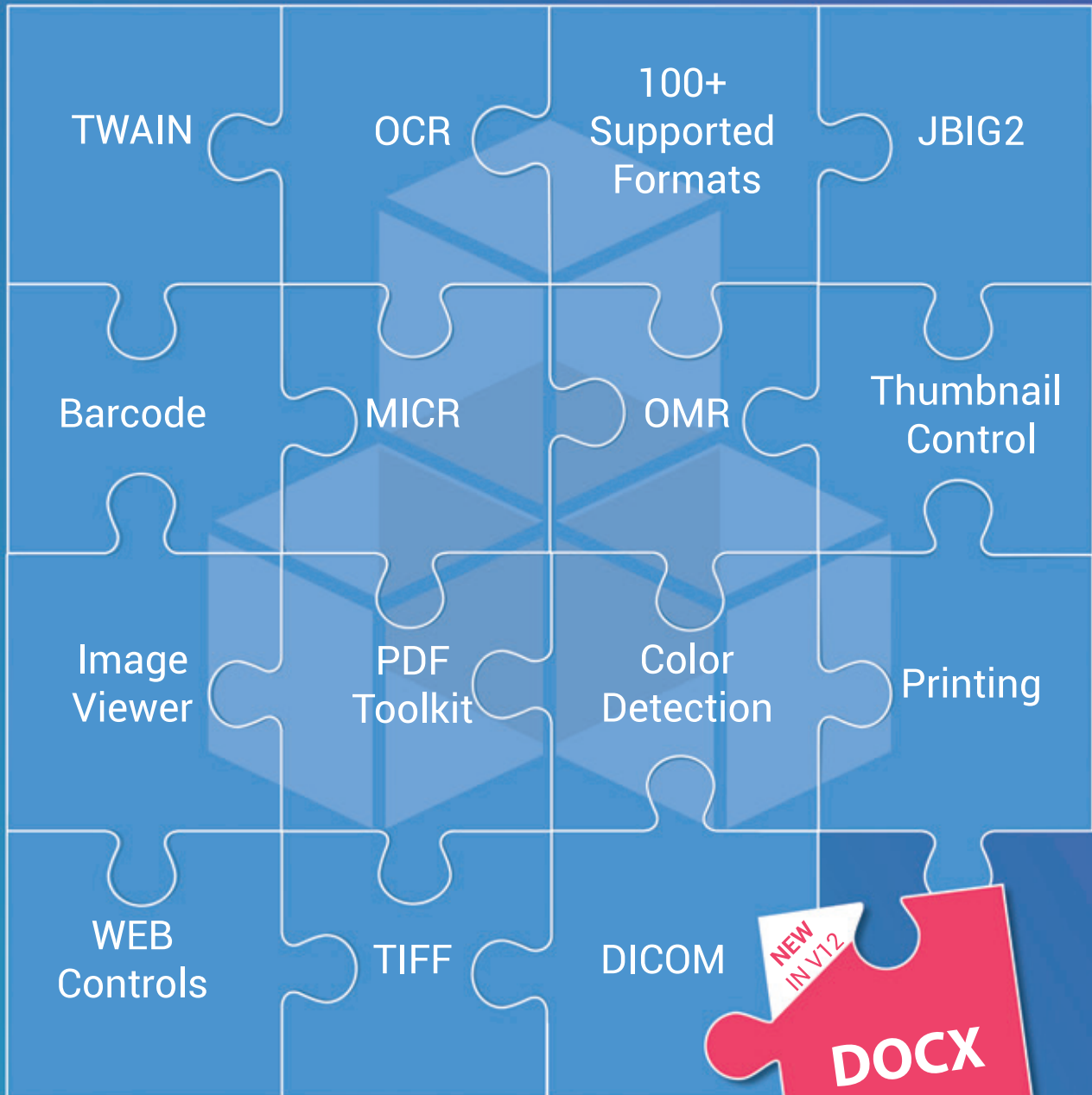
GdPicture.NET



12

100% ROYALTY FREE

Imaging SDK For **WinForms**, **WPF** And **Web** Development



Try **GdPicture.NET V12** for **Free** for 30 days

www.gdpicture.com

to randomly select the number of applicants in each trial. Behind the scenes, the Simulation method loops, calling a non-verbose version of the Select method, while keeping track of the number of times the best applicant was selected.

$\text{numSkip} = 10 / 2.71828 = 3$



Figure 2 The 1/e Algorithm for the Secretary Problem

The Select Method

The definition of the Select method begins as:

```
static int Select(double[] ratings)
{
    int numApplicants = ratings.Length;
    int numSkip = (int)(numApplicants / Math.E);
    int candidate = 0; // Best applicant seen
    double bestRating = ratings[0];
    double rating = 0.0;
    bool readyToHire = false;
    ...
}
```

For each applicant, the current applicant becomes the Candidate if the current applicant's rating is greater than the best (highest) rating seen.

The number of applicants is determined by the number of cells in the ratings array. You could make this explicit by passing an additional parameter N to Select. Variable numSkip is the number of initial applicants to skip over while tracking the best applicant so far. If you wanted to round instead of truncate, you could write:

```
int numSkip = (int)Math.Round(numApplicants / Math.E);
```

Variable candidate is the best applicant seen at any point in time during the hiring process and variable bestRating is the rating associated with the Candidate. The two variables are initialized to the values of the first applicant, rather than using dummy values such as -1 and -1.0, respectively.

Variable rating represents the rating of the current applicant, and is just for readability. Boolean variable readyToHire is false when processing the first numSkip-1 applicants, then set to true afterward.

The body of method Select is short and relatively simple:

```
...
for (int applicant = 0; applicant < numApplicants; ++applicant) {
    rating = ratings[applicant];
    if (applicant >= numSkip) readyToHire = true;

    if (rating > bestRating) {
        candidate = applicant;

        bestRating = rating;
        if (readyToHire == true) return candidate;
    }
}
return -1;
}
```

The method walks through each applicant in the ratings array. For each applicant, the current applicant becomes the Candidate

if the current applicant's rating is greater than the best (highest) rating seen. When Boolean variable readyToHire is false, Candidates are identified but not returned. When readyToHire is true, the first available Candidate found is returned, corresponding to an immediate hire in the Secretary Problem.

An alternative design is to use two loops instead of a single processing loop, where the first loop iterates over the applicants to skip, and the second loop identifies the first available Candidate:

```
// 1. prelim loop
for (int applicant = 0; applicant < numSkip; ++applicant)
{
    // Track best rating seen
}

// 2. hiring loop
for (int applicant = numSkip; applicant < numApplicants; ++applicant)
{
    // Return first Candidate found
}
```

It's possible that no new Candidate will appear after the first numSkip applicants. For example, suppose you have eight applicants with ratings (7.0, 6.0, 5.0, 4.0, 3.0, 2.0, 1.0, 0.0). Variable numSkip would be set to $8 / 2.71828 = 2$. After the first two applicants are interviewed, the best rating would be 7.0 and none of the remaining six applicants would be identified as a Candidate. Method Select returns a dummy value of -1 when no Candidate is found.

The Fisher-Yates algorithm is often used in machine learning and simulation programs to shuffle the values in a linear collection.

The Simulation Method

In high-level pseudo-code, the Simulation method is:

```
generate applicants / ratings
determine optimal rating
set number_times_best_applicant_selected = 0

loop numTrials times
    randomize order of applicants
    select an applicant using 1/e algorithm
    if selected applicant is optimal
        ++number_times_best_applicant_selected
    end-if
end-loop

return number_times_best_applicant_selected / numTrials
```

The definition of method `Simulate` begins with:

```
static double Simulation(int numApplicants, int numTrials)
{
    double[] ratings = new double[numApplicants];
    for (int i = 0; i < numApplicants; ++i)
        ratings[i] = (i * 1.0);

    double optimalRating = 1.0 * (numApplicants - 1);
    int numBestSelected = 0;

    ...
```

As described earlier, applicants and their ratings are stored in an array where the array index is the applicant ID, and the cell value is the rating. The value in cell 0 is set to 0.0, the value in cell 1 is set to 1.0, and so on. For example, if there are $N = 100$ applicants, the applicant IDs are 0 through 99 and the optimal rating is 99.0.

The main processing loop in method `Simulation` is:

```
for (int trial = 0; trial < numTrials; ++trial) {
    Shuffle(ratings);

    int selected = Select(ratings);
    if (selected == -1) // failed to select an applicant
        continue;

    double rating = ratings[selected];
    if (rating == optimalRating)
        ++numBestSelected;
}
```

A program-defined method named `Shuffle` rearranges the ratings in-place using the Fisher-Yates mini-algorithm:

```
static void Shuffle(double[] ratings)
{
    for (int i = 0; i < ratings.Length; ++i)
    {
        int ri = rnd.Next(i, ratings.Length); // random index in [i, N)
        double tmp = ratings[i];
        ratings[i] = ratings[ri];
        ratings[ri] = tmp;
    }
}
```

Many programming languages, for example Python and R, include a built-in shuffle method as part of their standard function library.

The Fisher-Yates algorithm is often used in machine learning and simulation programs to shuffle the values in a linear collection. Many programming languages, for example Python and R, include a built-in shuffle method as part of their standard function library. Here, the program-defined `Shuffle` method assumes the existence of a class-scope `Random` object:

```
class SecretaryProblemProgram
{
    static Random rnd = new Random(0);

    static void Main(string[] args)
    {
        ...
```

After the ratings are randomized, the processing loop in method `Simulation` calls the `Select` method to pick an applicant using the

1/e algorithm, and then the selected applicant is checked to see if they're the optimal applicant.

When checking for the optimal candidate, two floating-point values are compared for exact equality:

```
if (rating == optimalRating)
    ++numBestSelected;
```

Once you become familiar with the general structure of the Secretary Problem, you'll find quite a few scenarios where the key principles are relevant.

This approach is usually a bad idea because some floating-point values are stored only as approximations, to a fixed number of decimal points. The compare-for-exact-equality approach is OK in this particular example, but in general you'd want to compare for very closeness rather than exact equality.

Wrapping Up

The information presented in this article is based primarily on the 2003 research paper, "Analysis of Heuristic Solutions to the Best Choice Problem," by W. Stein, D. Seale and A. Rapoport. You can find the paper in PDF format in several places on the Internet. The paper presents a mathematical analysis of the 1/e rule and two additional rules.

The "candidate count" rule arbitrarily selects the n th Candidate. For example, if $n = 3$, the third Candidate encountered will be selected. In the example presented at the beginning of this article, where the applicants and ratings are (5.0, 2.0, 7.0, 1.0, 4.0, 0.0, 8.0, 3.0, 9.0, 6.0), the third Candidate is applicant 6 with a rating of 8.0, which coincidentally is the same as the 1/e rule. As it turns out, the candidate count rule is not very effective.

The "successive non-candidate" rule selects the first Candidate seen after observing k non-Candidates. For example, suppose $k = 3$. For the example applicants and ratings, the first k non-Candidates have ratings 2.0, 1.0 and 4.0. The next Candidate is applicant 6 with a rating of 8.0, which again is, coincidentally, the same as the 1/e rule. The successive non-candidate rule turns out to work quite well.

Once you become familiar with the general structure of the Secretary Problem, you'll find quite a few scenarios where the key principles are relevant. Whenever you're faced with a programming task that involves an uncertain stopping point, the ideas of the Secretary Problem might be useful. ■

DR. JAMES McCaffrey works for Microsoft Research in Redmond, Wash. He has worked on several Microsoft products including Internet Explorer and Bing. Dr. McCaffrey can be reached at jammc@microsoft.com.

THANKS to the following Microsoft technical experts who reviewed this article: Chris Lee and Kirk Olynyk



Command-Line Processing with .NET Core 1.0

In this month's Essential .NET column, I continue my investigation into the various features of .NET Core, this time with a fully released version (no longer in beta or a euphemistic release candidate). Specifically, I'm going to focus on its command-line utilities (which are found within the .NET Core Common library at github.com/aspnet/Common) and how to leverage them for parsing a command line. I confess I'm particularly excited to finally see command-line parsing support built into .NET Core as it's something I've wished for since the .NET Framework 1.0. I'm hoping that a .NET Core built-in library might help standardize, even if only a little, the command-line format/structure between programs. It isn't so important to me what the standard is as much as there be a convention that people default to, rather than everyone creating their own.

A Command-Line Convention

The bulk of the command-line functionality is found within the Microsoft.Extensions.CommandLineUtils NuGet package. Included in the assembly is a CommandLineApplication class that provides command-line parsing with support for short and long names for options, values (one or more) assigned with either a colon or an equals sign, and symbols like -? for help. Speaking of help, the class includes support for displaying the help text automatically. **Figure 1** shows some sample command lines that would be supported.

As described next, there are multiple argument types, one of which is called an "Argument." The overloading of the term argument to refer to the values specified on the command line versus the command-line configuration data can lead to significant ambiguity. Therefore, for the remainder of the article, I'll distinguish between a generic argument of any kind—specified after the executable name—and the argument type called "Argument" (title case) by the casing. Similarly, I'll distinguish the other argument types, Option and Command, using title case rather than the lowercase terms that generically refer to the argument. Please take note of this as it will be important throughout the remainder of the article.

Each of the argument types is described as follows:

- **Options:** Options are identified with a name, where the name is prefixed with either a single (-) or double dash (--). Option names are programmatically defined using templates and a template can include one or more of the following three designators: short

name, long name, symbol. In addition, an Option might have a value associated with it. For example, a template might be "-n | --name | -# <Full Name>," allowing the full name option to be identified by any of the three designators. (However, the template doesn't need all three designators.) Note that it's the use of a single or double dash that determines whether a short or long name is specified, regardless of the actual length of the name.

To associate a value with an option, you can use either a space or the assignment operator (=). -f=Inigo and -l Montoya, therefore, are both examples of specifying an option value.

If numbers are used in the template, they'll be part of the short or long names, not the symbol.

The overloading of the term argument to refer to the values specified on the command line versus the command-line configuration data can lead to significant ambiguity.

- **Arguments:** Arguments are identified by the order in which they appear rather than by a name. Thus, a value on the command line that isn't prefixed by an option name is an argument. Which argument the value corresponds to is based

Figure 1 Sample Command Lines

Options	Program.exe -f=Inigo, -l Montoya -hello -names Princess -names Buttercup	Option -f with value "Inigo" Option -l with value "Montoya" Option -hello with value "on" Option -names with values "Princess" and "Buttercup"
Commands with arguments	Program.exe "hello", "Inigo", "Montoya", "It", "is", "a", "pleasure", "to", "meet", "you."	Command "hello" Argument "Inigo" Argument "Montoya" Argument Greetings with values "It", "is", "a", "pleasure", "to", "meet", "you."
Symbols	Program.exe -?	Display help

Code download available at [IntelliTect.com/samples](https://intellitect.com/samples).

NEW! 1&1 MANAGED CLOUD HOSTING

The best of both worlds!

Scalable on-demand and flexible server resources combined with a new high-performance hosting package: the new **1&1 Managed Cloud Hosting** is here! Ideal for online projects which demand only the best availability, security and flexibility.

- ✓ **Dedicated resources**
- ✓ **20+ stack variations**
- ✓ **Managed by 1&1 experts**
- ✓ **Flexible scalability**
- ✓ **Deployed <1 minute**



Trusted Performance.
Intel® Xeon® processors.

**TRY FREE
FOR ONE MONTH!***



☎ 1-844-296-2059



1and1.com

*1&1 Managed Cloud Hosting first month free, then \$9.99/month. You may cancel any time during the first month. No set up fee. © 1&1 Internet Inc. 2016 All rights reserved. 1&1 and the 1&1 logo are trademarks of 1&1 Internet SE, all other trademarks are the property of their respective owners. 1&1 Internet Inc., 701 Lee Road, Chesterbrook, PA 19087

on the order in which it appears (Options and Commands are excluded in the count).

- **Commands:** Commands provide a grouping of arguments and options. For example, you can have a command name “hello” followed by a combination of Arguments and Options (or even sub-Commands). Commands are identified by a configured keyword, the command name, that groups all the values following the command name to be part of that Command’s definition.

Configuring the Command Line

Programming the command line after referencing the .NET Core Microsoft.Extensions.CommandLineUtils begins with the CommandLineApplication class. With this class you’re able to configure each Command, Option and Argument. When instantiating the CommandLineApplication, the constructor has an optional Boolean value that configures the command line to throw an exception (the default) if an argument appears that hasn’t been specifically configured.

Given an instance of the CommandLineApplication, you configure arguments using the Option, Argument, and Command methods. Imagine, for example, you want to support a command-line syntax as follows, where items in square brackets are optional and those in angle brackets are user-specified values or arguments:

```
Program.exe <-g|--greeting>-$ <greeting>> [name <fullname>] [-?|-h|--help]
[-u|--uppercase]
```

Figure 2 configures the basic parsing capability.

One elegant way to
save all the command-line data
is to place it into a separate
class that’s decorated with the
attributes from the ASP.NET
Scaffolding repo.

It Starts with CommandLineApplication

To begin, I instantiate the CommandLineApplication, specifying whether the command-line parsing will be strict—throwOnUnexpectedArg is true—or relaxed. If I specify to throw an exception when an argument is unexpected, all arguments will have to be explicitly configured. Alternatively, if throwOnUnexpectedArg is false, then any arguments that aren’t recognized by the configuration will be stored into the CommandLineApplication.RemainingArguments field.

Configuring a Command and Its Argument

The next step in **Figure 2** is configuring the “name” Command. The keyword that will identify the command within a list of arguments is the first parameter of the Command function—name.

The second parameter is an Action<CommandLineApplication> delegate called configuration, into which all sub-arguments of the name Command are configured. In this case, there’s only one, an Argument of type CommandArgument with the variable name of “greeting.” However, it’s entirely possible to add additional Arguments,

If I specify to throw an
exception when an argument is
unexpected, all arguments will
have to be explicitly configured.

Options and even sub-Commands within the configuration delegate. Furthermore, the target parameter of the delegate, a CommandLineApplication, has a Parent property that points back to commandLineArgument—the parent CommandLineArgument of target under which the name Command is configured.

Figure 2 Configuring the Command Line

```
public static void Main(params string[] args)
{
    // Program.exe <-g|--greeting>-$ <greeting>> [name <fullname>]
    // [-?|-h|--help] [-u|--uppercase]

    CommandLineApplication commandLineApplication =
        new CommandLineApplication(throwOnUnexpectedArg: false);

    CommandArgument names = null;
    commandLineApplication.Command("name",
        (target) =>
        {
            names = target.Argument(
                "fullname",
                "Enter the full name of the person to be greeted.",
                multipleValues: true);
        });

    CommandOption greeting = commandLineApplication.Option(
        "-$|-g |--greeting <greeting>",
        "The greeting to display. The greeting supports"
        + " a format string where {fullname} will be "
        + "substituted with the full name.",
        CommandOptionType.SingleValue);

    CommandOption uppercase = commandLineApplication.Option(
        "-u |--uppercase", "Display the greeting in uppercase.",
        CommandOptionType.NoValue);

    commandLineApplication.HelpOption("-? | -h | --help");

    commandLineApplication.OnExecute(() =>
    {
        if (greeting.HasValue())
        {
            Greet(greeting.Value(), names.Values, uppercase.HasValue());
        }
        return 0;
    });

    commandLineApplication.Execute(args);
}

private static void Greet(
    string greeting, IEnumerable<string> values, bool useUppercase)
{
    Console.WriteLine(greeting);
}
```




ORLANDO
ROYAL PACIFIC RESORT AT
UNIVERSAL ORLANDO

**DEC
5-9**

**A NEW CONFERENCE FOR
SOFTWARE DEVELOPERS**

APPDEV TRENDS

ENTERPRISE FOCUSED. CODE DRIVEN.

Powering Enterprise Development

App Dev Trends, brought to you by ADTmag.com, is a new technology conference focused on the makers and maintainers of the purpose-designed software that Power organizations in virtually every industry in the world—in other words, enterprise software professionals! You Power your company. It's our job to Power you!

This event is for:

- down-in-the-trenches developers
- team leaders
- entire software development teams

Track topics include:

- Agile
- Cloud
- Mobility
- Java
- Containerization
- Continuous Integration



**REGISTER BY OCTOBER 5
AND SAVE \$400!**



Use promo code ADTSEP1

Scan the QR code to register
or for more event details.

A Part of Live! 360: The Ultimate Education Destination

6 Great Conferences, 1 Great Price

Visual Studio **LIVE!**

ModernApps **LIVE!**

TECHMENTOR

Office &
SharePoint **LIVE!**

SQL Server **LIVE!**

**APPDEV
TRENDS** **NEW!**

APPDEVTRENDS.COM



PRODUCED BY
1105 MEDIA

Notice in configuring the names Argument I specifically identify that it will support multipleValues. In so doing, I allow more than one value to be specified—multiple names in this case. Each of these values appears following the “name” argument identifier until another argument or option identifier appears. The first two parameters of the Argument function are name, referring to the Argument’s name so you can identify it from a list of Arguments, and description.

One last thing to point out in the name Command configuration is the fact that you need to save the return from the Argument function (and the Option function if there’s any). This is necessary so you can later retrieve the arguments associated with the names Argument. Without saving a reference, you end up having to search through the CommandLineApplication.Commands[0].Arguments collection in order to retrieve the Argument data.

One elegant way to save all the command-line data is to place it into a separate class that’s decorated with the attributes from the ASP.NET Scaffolding repo (github.com/aspnet/Scaffolding), specifically the `src/Microsoft.VisualStudio.Web.CodeGeneration.Core/CommandLine` folder. For more information, see “Implementing a Command-Line Class with .NET Core” (bit.ly/296SluA).

Configuring an Option

The next argument configured in **Figure 2** is the greeting Option, which is of type `CommandOption`. Configuration of an Option is done via the `Option` function, where the first parameter is a string parameter called template. Notice that you can specify three different names (for example, `-$`, `-g`, and `-greeting`) for the option and each of these will be used to identify the option from the list of arguments. Also, a template can optionally specify a value associated with it by means of an angle-bracketed name following the option identifiers. After the description parameter, the `Option` function includes a required `CommandOptionType` parameter. This option identifies:

1. Whether any value may be specified after the option identifier. If a `CommandOptionType` of `NoValue` is specified, then the `CommandOption.Value` function will be set to “on” if the option appears within the list of arguments. The value “on” is returned even if a different value is specified following the option identifier and, in fact, whether a value is specified. To see an example, review the uppercase option in **Figure 2**.
2. Alternatively, if the `CommandOptionType` is `SingleValue` and the option identifier is specified but no value appears, a `CommandParsingException` will be thrown identifying the option was not identified—because it didn’t match the template. In other words, `SingleValue` provides a means of checking that the value is provided, assuming the option identifier appears at all.

Figure 3 ShowHelp Display Output

```
Usage: [options] [command]
Options:
  -$|-g|--greeting <greeting> The greeting to display. The greeting supports
a format string where {fullname} will be substituted with the full name.
  -u|--uppercase                Display the greeting in uppercase.
  -?|-h|--help                 Show help information
Commands:
  name
Use " [command] --help" for more information about a command.
```

3. Last, you can provide a `CommandOptionType` of `MultipleValue`. Unlike the multiple values associated with a command, however, multiple values in the case of an option allow the same option to be specified multiple times. For example, `program.exe -name Inigo -name Montoya`.

Notice that none of the configuration options will configure so the option is required. And, in fact, the same is true for an argument. To error out if a value isn’t specified, you need to check whether the `HasValue` function reports an error if it returns false. In the case of a `CommandArgument`, the `Value` property will return null if no value is specified. To report the error, consider displaying an error message followed by the help text so users have more information about what they need to do to correct the problem.

Another important behavior of the `CommandLineApplication` parsing mechanism is that it’s case-sensitive. And, in fact, at this time there’s no easy configuration option that allows you to make it case-insensitive. Therefore, you’ll need to change the case of the actual arguments passed into the `CommandLineApplication` (via the `Execute` method, as I’ll describe shortly) beforehand to achieve case insensitivity. (Alternatively, you could try submitting a pull request at github.com/aspnet/Common to enable this option.)

Display Help and Version

Built into the `CommandLineApplication` is a `ShowHelp` function that displays the help text associated with the command-line configuration automatically. For example, **Figure 3** shows the `ShowHelp` output for **Figure 2**.

Another important behavior of the `CommandLineApplication` parsing mechanism is that it’s case-sensitive.

Unfortunately, the help displayed doesn’t identify whether an option or command is, in fact, optional. In other words, the help text assumes and displays (via square brackets) that all options and commands are optional.

Although you can call `ShowHelp` explicitly, for example when handling a custom command-line error, it will be automatically invoked whenever an argument matching the `HelpOption` template is specified. And, the `HelpOption` template is specified via an argument to the `CommandLineApplication.HelpOption` method.

Similarly, there’s a `ShowVersion` method for displaying the version of your application. Like `ShowHelp`, it’s configured via one of two methods:

```
public CommandOption VersionOption(
    string template, string shortFormVersion, string longFormVersion = null).
public CommandOption VersionOption(
    string template, Func<string> shortFormVersionGetter,
    Func<string> longFormVersionGetter = null)
```

Notice that both methods require the version information you wish to display to be specified in the call to `VersionOption`.

Extreme Performance Linear Scalability

For .NET Applications

Open Source (Apache 2.0)



Distributed Cache

- In-Memory App Data Caching
- ASP.NET Sessions & View State
- Runtime Data Sharing



NoSQL Database

- Schema-less JSON Documents
- Data Consistency & Replication
- Powerful SQL Support

100% Native .NET



Parsing and Reading the Command-Line Data

So far I've reviewed in detail how to configure the `CommandLineApplication`, but I haven't yet discussed the ever-so-critical process of triggering command-line parsing, or what will happen immediately following the parse invocation.

To trigger command-line parsing you need to invoke the `CommandLineApplication.Execute` function and pass the list of arguments specified on the command line. In **Figure 1**, the arguments are specified in the `args` parameter of `Main` so they're passed into the `Execute` function directly (remember to first handle casing if case sensitivity isn't desirable). It's the `Execute` method that sets the command-line data associated with each `Argument` and `Option` that's configured.

To trigger command-line parsing
you need to invoke the
`CommandLineApplication.Execute`
function and pass the list of
arguments specified on the
command line.

Note that `CommandLineApplication` includes an `OnExecute(Func<int> invoke)` function into which you can pass a `Func<int>` delegate that will execute automatically once the parsing completes. In **Figure 2**, the `OnExecute` method takes a simple delegate that checks that the `greet` command was specified before invoking a `Greet` function.

Note also that the `int` returned from the `invoke` delegate is designed as a means of specifying a return value from `Main`. And, in fact, whatever value is returned from `invoke` will correspond to the return from `Execute`. Moreover, because parsing is considered a relatively slow operation (I suppose it's all relative), `Execute` supports an overload that takes a `Func<Task<int>>`, thus enabling an asynchronous invocation of the command-line parsing.

Guidelines: Commands, Arguments and Options

Given the three command types available, it's worth quickly reviewing which to use when.

Do use `Commands` when semantically identifying an action such as `compile`, `import` or `backup`.

Do use `Options` to enable configuration information to either the program as a whole or a specific command.

Favor a verb for the name of a command and an adjective or noun for the name of an option (such as `-color`, `-parallel`, `-projectname`).

Regardless of which argument type you configure, consider the following guidelines:

Do review the case of argument identifier names. It could be very confusing for a user who specifies `-FullName` or `-fullname` when the command line is looking for a different case.

Do write tests for the command-line parsing. Methods like `Execute` and `OnExecute` make this relatively easy to do.

Do use `Arguments` when identifying particular arguments by name is cumbersome or when multiple values are allowed but prefixing each one with an option identifier is unwieldy.

Consider leveraging `IntelliTect.AssertConsole` (itl.tc/CommandLineUtils) for redirecting console input and output in order to inject and capture the console so it can be tested.

There's one possible drawback to using the .NET Core `CommandLineUtils` and that's that they're English-based and not localized. Display text such as that found in `ShowHelp` (along with exception messages that generally aren't localized) are all in English. Normally, this might not be a problem but because a command line is part of an application's interface with the user, there are likely to be scenarios where English-only is unacceptable. For this reason:

Consider writing custom functions for `ShowHelp` and `ShowHint` if localization is important.

Do check `CommandLineApplication.RemainingArguments` when the `CommandLineApplication` is configured to not throw exceptions (`throwOnUnexpectedArg = false`).

Wrapping Up

Over the last three years the .NET Framework has gone through some major transitions:

- It now has cross-platform support, including support for iOS, Android and Linux—Wow!!
- It has migrated from a secret, proprietary approach toward development to an entirely open—as in open source—module.
- There has been a significant refactoring of the BCL APIs for the .NET Standard Library into a highly modular (cross) platform that can be leveraged over the vast range of application types out there, whether Software as a Service, mobile, on-premises, Internet of Things, desktop and more.
- There has been a rebirth of .NET, following the Windows 8 era where it was ignored with very little strategy or road-map of note.

All that's to say, if you haven't yet started to dive into the new .NET Core 1.0, now is a great time to do so, giving you the longest time span in which to amortize the learning curve. In other words, if you're contemplating upgrading to it from earlier versions, do so now. Chances are good you'll be upgrading at some point, and the sooner you do, the sooner you can take advantage of its new features. ■

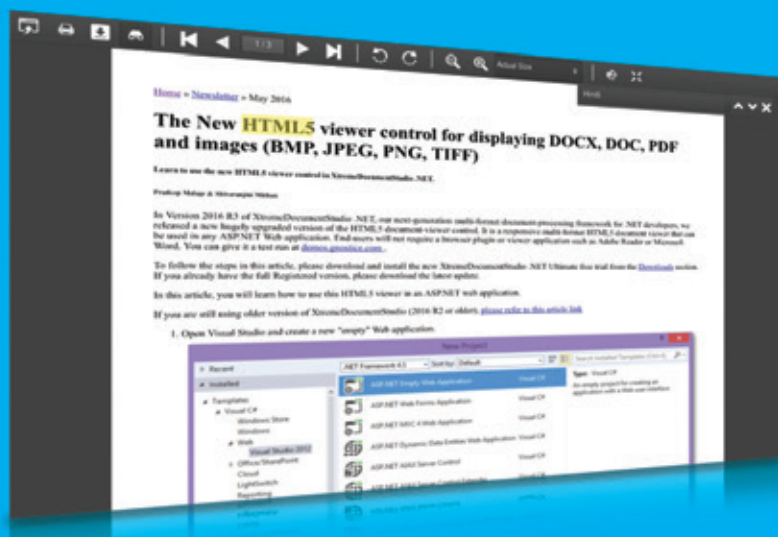
MARK MICHAELIS is founder of *IntelliTect*, where he serves as its chief technical architect and trainer. For nearly two decades he has been a Microsoft MVP, and a Microsoft Regional Director since 2007. Michaelis serves on several Microsoft software design review teams, including C#, Microsoft Azure, SharePoint and Visual Studio ALM. He speaks at developer conferences and has written numerous books including his most recent, "Essential C# 6.0 (5th Edition)" (itl.tc/EssentialCSharp). Contact him on Facebook at facebook.com/Mark.Michaelis, on his blog at IntelliTect.com/Mark, on Twitter: [@markmichaelis](https://twitter.com/markmichaelis) or via e-mail at mark@IntelliTect.com.

THANKS to the following *IntelliTect* technical experts for reviewing this article: Phil Spokas and Michael Stokesbary

Document Technology for Everybody



Introducing The New Interactive Multi-Format HTML5 Document Viewer



- Responsive HTML5 control, automatically adjusts for Mobile, Desktop and Pad/Tablets.
- 100% Independent. No browser plug-ins required. No ActiveX. No Office Automation.
- Single viewer for PDF, Office documents, text files and Images.
- Text Search, On-the-fly OCR on images, PDF form-filling, and more...
- Full-fledged client-side JavaScript to configure and perform all operations.
- TypeScript Support.
- Simple, straightforward licensing.



XtremeDocumentStudio
.NET



StarDocs API
Server



XtremeDocumentStudio
Java



XtremeDocumentStudio
Delphi

Orlando
2016

ROYAL PACIFIC RESORT AT
UNIVERSAL ORLANDO

DEC
5-9



Visual Studio® LIVE!

EXPERT SOLUTIONS FOR .NET DEVELOPERS

Journey into Code

Join us as we journey into real-world, practical education and training on the Microsoft Platform. Visual Studio Live! (VSLive!™) returns to warm, sunny Orlando for the conference more developers rely on to expand their .NET skills and the ability to build better applications.



twitter.com/live360
@live360



facebook.com
Search "Live 360"



linkedin.com
Join the "Live! 360" group!



EVENT PARTNERS



PLATINUM SPONSORS



GOLD SPONSORS



SUPPORTED BY





TECH EVENTS WITH PERSPECTIVE

6 Great Conferences 1 Great Price

Visual Studio Live! Orlando is part of Live! 360, the Ultimate Education Destination. This means you'll have access to five (5) other co-located events at no additional cost:

SQL Server **LIVE!**
TRAINING FOR DBAS AND IT PROS

TECHMENTOR
IN-DEPTH TRAINING FOR IT PROS

Office & SharePoint **LIVE!**
ON-PREMISE, CLOUD & CROSS-PLATFORM TRAINING

ModernApps **LIVE!**
MOBILE, CROSS-DEVICE & CLOUD DEVELOPMENT

NEW! **APPDEV TRENDS**
ENTERPRISE FOCUSED. CODE DRIVEN.

Six (6) events and hundreds of sessions to choose from - mix and match sessions to create your own, custom event line-up - it's like no other conference available today!

Whether you are an

- Engineer
- Developer
- Programmer
- Software Architect
- Software Designer

You will walk away from this event having expanded your .NET skills and the ability to build better applications.

REGISTER TODAY AND **SAVE \$400!**



Use promo code
ORLSEP4 by October 5

Scan the QR code to
register or for more
event details.

TURN THE PAGE FOR
MORE EVENT DETAILS



Redmond
MAGAZINE

VIRTUALIZATION
REVIEW

ADT MAG
MAGAZINE

Visual Studio
MAGAZINE

PRODUCED BY

1105 MEDIA

VSLIVE.COM/ORLANDO

Check Out the Additional Sessions for Devs, IT Pros, & DBAs at Live! 360



Office & SharePoint LIVE!

ON-PREMISE, CLOUD & CROSS-PLATFORM TRAINING

Office & SharePoint Live!
features 12+ developer
sessions, including:

- Workshop: A Beginner's Guide to Client Side Development in SharePoint - *Mark Rackley*
- Become a Developer Hero by Building Office Add-ins - *Bill Ayres*
- Utilizing jQuery in SharePoint - Get More Done Faster - *Mark Rackley*
- Using the Office UI Fabric - *Paul Schaefflein*
- Enterprise JavaScript Development Patterns - *Rob Windsor*
- Leveraging Angular2 to Build Office Add-ins - *Andrew Connell*
- Webhooks in Office 365 - *Paul Schaefflein*



SQL Server LIVE!

TRAINING FOR DBAs AND IT PROS

SQL Server Live! features 15+
developer sessions, including:

- What's New in SQL Server 2016 - *Leonard Lobel*
- Powerful T-SQL Improvements that Reduce Query Complexity - *Hugo Kornelius*
- Implementing Data Protection and Security in SQL Server 2016 - *Steve Jones*
- Welcome To The 2016 Query Store! - *Janis Griffin*
- Workshop: Big Data, Analytics and NoSQL: Everything You Wanted to Learn But Were Afraid to Ask - *Andrew Brust*



TECHMENTOR

IN-DEPTH TRAINING FOR IT PROS

TechMentor features IT Pro
and DBA sessions, including:

- Workshop: 67 VMware vSphere Tricks That'll Pay for This Conference! - *Greg Shields*
- Secure Access Everywhere! Implementing DirectAccess in Windows Server 2016 - *Richard Hicks*
- Getting Started with Nano Server - *Jeffery Hicks*
- Creating Class-Based PowerShell Tools - *Jeffery Hicks*
- Harvesting the Web: Using PowerShell to Scrape Screens, Exploit Web Services, and Save Time - *Mark Minasi*
- PowerShell Unplugged: Stump Don - *Don Jones*
- Facing Increasing Malware Threats and a Growing Trend of BYO with a New Approach of PC Security - *Yung Chou*



ALM / DevOps	Cloud Computing	Mobile Client	Software Practices	Visual Studio / .NET Framework
--------------	-----------------	---------------	--------------------	--------------------------------

START TIME	END TIME		
5:00 PM	8:00 PM	Pre-Conference Registration - Royal Pacific Resort Conference Center	
6:00 PM	9:00 PM	Dine-A-Round Dinner @ Universal CityWalk	
START TIME	END TIME		
8:00 AM	5:00 PM	VSM01 Workshop: Distributed Cross-Platform Application Architecture - Rockford Lhotka & Jason Bock	
12:00 PM	1:00 PM	Lunch	
1:00 PM	5:00 PM	VSM1 Workshop Continues	
5:00 PM	6:00 PM	EXPO Preview	
6:00 PM	7:00 PM	Live! 360 Keynote: To Be Announced	
START TIME	END TIME		
8:00 AM	9:00 AM	Visual Studio Live! Keynote: To Be Announced	
9:00 AM	9:30 AM	Networking Break • Visit the EXPO	
9:30 AM	10:45 AM	VST01 Building Applications with ASP.NET Core - Scott Allen	VST02 Busy .NET Developer's Guide to Swift - Ted Neward
11:00 AM	12:15 PM	VST05 Richer MVC Sites with Knockout JS - Miguel Castro	VST06 Busy .NET Developer's Guide to Native iOS - Ted Neward
12:15 PM	2:00 PM	Lunch • Visit the EXPO	
2:00 PM	3:15 PM	VST09 WCF & Web API: Can We All Just Get Along?!? - Miguel Castro	VST10 Creating Great Looking Android Applications Using Material Design - Kevin Ford
3:15 PM	4:15 PM	Networking Break • Visit the EXPO	
4:15 PM	5:30 PM	VST13 Busy Developer's Guide to Chrome Development - Ted Neward	VST14 Using Visual Studio Tools for Apache Cordova to Create MultiPlatform Applications - Kevin Ford
5:30 PM	7:30 PM	Exhibitor Reception	
START TIME	END TIME		
8:00 AM	9:15 AM	VSW01 Moving from Angular 1 to Angular 2 - Ben Dewey	VSW02 The Future of Mobile Application Search - James Montemagno
9:30 AM	10:45 AM	VSW05 Getting Started with Aurelia - Brian Noyes	VSW06 Building Connected and Disconnected Mobile Applications - James Montemagno
10:45 AM	11:15 AM	Networking Break • Visit the EXPO	
11:15 AM	12:15 PM	Live! 360 Keynote: To Be Announced	
12:15 PM	1:45 PM	Birds-of-a-Feather Lunch • Visit the EXPO	
1:45 PM	3:00 PM	VSW09 Living in a Command Line Web Development World (NPM, Bower, Gulp, and More) - Ben Dewey	VSW10 Understanding the Windows Desktop App Development Landscape - Brian Noyes
3:00 PM	4:00 PM	Networking Break • Visit the EXPO • Expo Raffle @ 3:30 p.m.	
4:00 PM	5:15 PM	VSW13 Securing Client JavaScript Apps - Brian Noyes	VSW14 Let's Write a Windows 10 App: A Basic Introduction to Universal Apps - Billy Hollis
8:00 PM	10:00 PM	Live! 360 Dessert Luau - Wantilan Pavilion	
START TIME	END TIME		
8:00 AM	9:15 AM	VSH01 Build Real-Time Websites and Apps with SignalR - Rachel Appel	VSH02 Cognitive Services: Building Smart Applications with Computer Vision - Nick Landry
9:30 AM	10:45 AM	VSH05 HTTP/2: What You Need to Know - Robert Boedigheimer	VSH06 Building Business Apps on the Universal Windows Platform - Billy Hollis
11:00 AM	12:15 PM	VSH09 TypeScript and ES2015 JumpStart - John Papa	VSH10 A Developers Introduction to HoloLens - Billy Hollis & Brian Randell
12:15 PM	1:30 PM	Lunch on the Lanai	
1:30 PM	2:45 PM	VSH13 All Your Tests Are Belong To Us - Rachel Appel	VSH14 Developing Awesome 3D Apps with Unity and C# - Adam Tuliper
3:00 PM	4:15 PM	VSH17 SASS and CSS for Developers - Robert Boedigheimer	VSH18 From Oculus to HoloLens: Building Virtual & Mixed Reality Apps & Games - Nick Landry
4:30 PM	5:30 PM	Live! 360 Conference Wrap-Up - Pacifica 6 - Andrew Brust (Moderator),	
START TIME	END TIME		
8:00 AM	5:00 PM	VSF01 Workshop: Angular 2 Bootcamp - John Papa	
12:00 PM	1:00 PM	Lunch	
1:00 PM	5:00 PM	VSF01 Session Continues	

Speakers and sessions subject to change

AGENDAS AT-A-GLANCE

Visual Studio **LIVE!**
EXPERT SOLUTIONS FOR .NET DEVELOPERSModernApps **LIVE!**
MOBILE, CROSS-DEVICE & CLOUD DEVELOPMENTPresented in
Partnership with **Magenic****APPDEV
TRENDS**
ENTERPRISE FOCUSED. CODE DRIVEN.

Web Client

Web Server

Windows Client

Modern Apps Live!

Agile

Containerization

Continuous
Integration

Java

Mobile

Cloud

Pre-Conference: Sunday, December 4, 2016

Pre-Conference Workshops: Monday, December 5, 2016

VSM02 Workshop: Service Oriented Technologies - Designing, Developing, & Implementing WCF and the Web API - *Miguel Castro***VSM03** Workshop: DevOps in a Day - *Brian Randell***MAM01** Workshop: Building Modern Mobile Apps - *Brent Edwards & Kevin Ford***ADM01** Workshop: Building Teams - *Steve Green***ADM02** Workshop: One Codebase to Rule Them All: Xamarin - *Fabian Williams***VSM2** Workshop Continues**VSM3** Workshop Continues**MAM01** Workshop Continues**ADM01** Workshop Continues**ADM02** Workshop Continues

Day 1: Tuesday, December 6, 2016

VST03 What's New in Azure v2 - *Eric D. Boyd***VST04** Real World Scrum with Team Foundation Server 2015 & Visual Studio Team Services - *Benjamin Day***MAT01** Modern App Development: Transform How You Build Web and Mobile Software - *Rockford Lhotka***ADT01** Hacking Technical Debt - *Steve Green***ADT02** Java 8 Lambdas and the Streaming API - *Michael Remijan***VST07** Overview of Power Apps - *Nick Pinheiro***VST08** Get Good at DevOps: Feature Flag Deployments with ASP.NET, WebAPI, & JavaScript - *Benjamin Day***MAT02** Architecture: The Key to Modern App Success - *Brent Edwards***ADT03** Are You A SOLID Coder? - *Steve Green***ADT04** PrimeFaces 5: Modern UI Widgets for Java EE - *Kito Mann***VST11** Introduction to Next Generation of Azure PaaS - Service Fabric and Containers - *Vishwas Lele***VST12** To Be Announced**MAT03** Manage Distributed Teams with Visual Studio Team Services and Git - *Brian Randell***ADT05** Agile Architecture - *Steve Green***ADT06** Full Stack Java with JSweet, Angular 2, PrimeNG, and JAX-RS - *Kito Mann***VST15** Cloud Oriented Programming - *Vishwas Lele***VST16** Bringing DevOps to the Database - *Steve Jones***MAT04** Focus on the User Experience #FTW - *Anthony Handley***ADT07** Crafting Innovation - *Steve Green***ADT08** Who's Taking Out the Garbage? How Garbage Collection Works in the VM - *Kito Mann*

Day 2: Wednesday, December 7, 2016

VSW03 Managing Enterprise and Consumer Identity with Azure Active Directory - *Nick Pinheiro***VSW04** Improving Performance in .NET Applications - *Jason Bock***MAW01** DevOps, Continuous Integration, the Cloud, and Docker - *Dan Nordquist***ADW01** Stop Killing Requirements! - *Melissa Green***ADW02** Migrating Customers to Microsoft Azure: Lessons Learned From the Field - *Ido Flatow***VSW07** Practical Internet of Things for the Microsoft Developer - *Eric D. Boyd***VSW08** I'll Get Back to You: Understanding Task, Await, and Asynchronous Methods - *Jeremy Clark***MAW02** Mobile Panel - *Kevin Ford, Rockford Lhotka, James Montemagno, & Jordan Matthiesen***ADW03** Meeting-Free Software Development in Distributed Teams - *Yegor Bugayenk***ADW04** The Essentials of Building Cloud-Based Web Apps with Azure - *Ido Flatow***VSW11** To Be Announced**VSW12** Learn to Love Lambdas (and LINQ, Too) - *Jeremy Clark***MAW03** C# Everywhere: How CSLA .NET Enables Amazing Cross-Platform Code Reuse - *Rockford Lhotka***ADW05** Introduction to Microsoft Office Graph - *Fabian Williams***ADW06** Building IoT and Big Data Solutions on Azure - *Ido Flatow***VSW15** ARM Yourself for Azure Success - *Esteban Garcia***VSW16** Continuous Delivery on Azure: A/B Testing, Canary Releases, and Dark Launching - *Marcel de Vries***MAW04** Coding for Quality and Maintainability - *Jason Bock***ADW07** As You Think About Azure Databases, Think About DocumentDB - *Fabian Williams***ADW08** Where Does JavaScript Belong in the App Store? - *Jordan Matthiesen*

Day 3: Thursday, December 8, 2016

VSH03 C# Best Practices - *Scott Allen***VSH04** Application Insights: Measure Your Way to Success - *Esteban Garcia***MAH01** Modern Mobile Development: Build a Single App For iOS & Android with Xamarin Forms - *Kevin Ford***ADH01** From VMs to Containers: Introducing Docker Containers for Linux and Windows Server - *Ido Flatow***ADH02** Continuous Testing in a DevOps World - *Wayne Ariola***VSH07** Debugging Your Way Through .NET with Visual Studio 2015 - *Ido Flatow***VSH08** The Ultimate Intro to Docker for Developers - *Adam Tuliper***MAH02** Universal Windows Development: UWP for PC, Tablet & Phone - *Brent Edwards***ADH03** CQRS 2.0 - Commands, Actors, and Events...Oh My! - *David Hoerster***ADH04** Microservices as Chat Bots Are the Future - *Yegor Bugayenk***VSH11** Exploring Microservices in a Microsoft Landscape - *Marcel de Vries***VSH12** Automated UI Testing for iOS and Android Mobile Apps - *James Montemagno***MAH03** Modern Web Development: ASP.NET MVC and Web API - *Allen Conway***ADH05** The Curious Case for the Immutable Object - *David Hoerster***ADH06** Continuous Integration May Have Negative Effects - *Yegor Bugayenk***VSH15** Unit Testing Makes Me Faster: Convincing Your Boss, Your Co-Workers, and Yourself - *Jeremy Clark***VSH16** Writing Maintainable, X-Browser Automated Tests - *Marcel de Vries***MAH04** Modern Web Development: Building a Smart Web Client with TypeScript and Angular2 - *Allen Conway***ADH07** To Be Announced**ADH08** Mobile DevOps Demystified with Xamarin, VSTS and HockeyApp - *Roy Cornelissen***VSH19** User Experience Case Studies - Good and Bad - *Billy Hollis***VSH20** Debugging the Web with Fiddler - *Ido Flatow***MAH05** Using All That Data: Power BI to the Rescue - *Scott Diehl***ADH09** Get Started with Microsoft PowerApps - *Fabian Williams***ADH10** Overcoming the Challenges of Mobile Development in the Enterprise - *Roy Cornelissen*

Andrew Connell, Don Jones, Rockford Lhotka, Matthew McDermott, Brian Randell, & John K. Waters

Post-Conference Workshops: Friday, December 9, 2016

VSF02 Workshop: Building Modern Web Apps with Azure - *Eric D. Boyd & Brian Randell***MAF01** Workshop: Modern App Deep Dive: Xamarin, Responsive Web, UWP, CSLA .NET - *Jason Bock, Allen Conway, Brent Edwards & Kevin Ford***ADF01** Workshop: To Be Announced**VSF02** Session Continues**MAF01** Session Continues**ADF01** Session Continues



Building a City-Wide Wi-Fi Scanner with UWP and Azure

In my July column (msdn.com/magazine/mt736460), I covered how to write a Universal Windows Platform (UWP) app to scan Wi-Fi networks. At the end of the article, I provided the solution to create a CSV text string. I then briefly mentioned what could be done with Power BI in terms of data visualization. However, the manual process of copying and pasting text data from the UI prevents the solution from scaling beyond a single user. Furthermore, the design goal is to run the app on an IoT device without any type of display. This means that the UWP app must send data to the cloud without any UI. This is also known as a “headless” solution. The plan entails installing numerous devices around a city to scan for Wi-Fi Networks.

First, you must set up a cloud service to take in all this data. Second, you must modify the existing UWP app's code to send data to that cloud service. Finally, you want to visualize this data in Power BI.

As the term implies,
the Internet of Things is about
the Internet *and* the things
connected to the Internet.

The Cloud and the Internet of Things

The cloud is a marvelous innovation. It offers effectively limitless storage and compute to even the lowest-powered devices. All that's needed is a connection to the Internet. Often when the Internet of

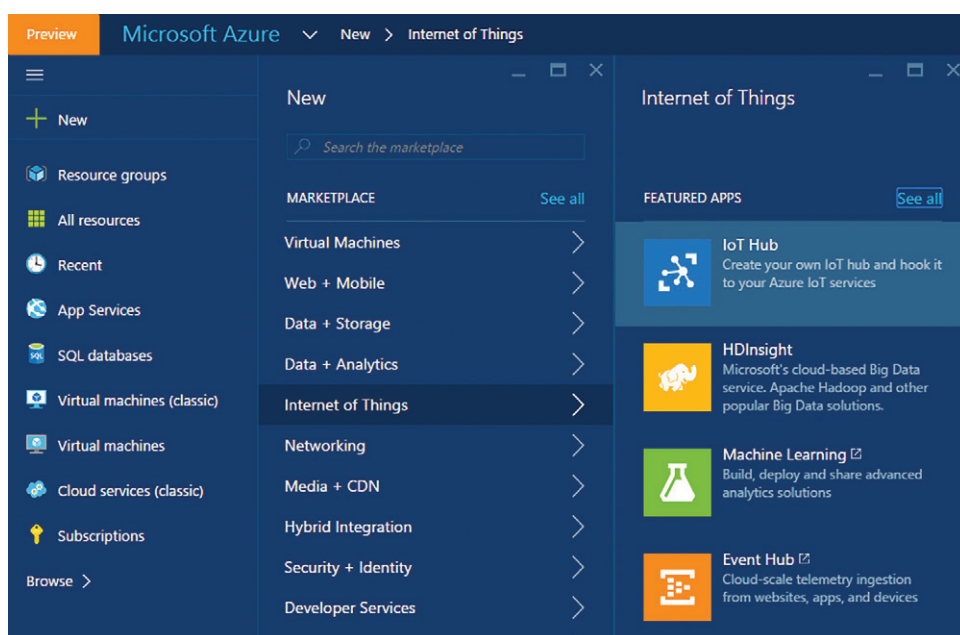


Figure 1 Creating an IoT Hub in the Azure Portal

Things (IoT) is discussed, cloud services dominate the conversation, rather than the actual devices. Why is that? Well, simply put, many of the devices connected to the Internet need a central place from which to transmit data and receive notifications. Much of the world-changing, paradigm-shifting outcomes are the result of data being crunched and analyzed in the cloud. Additionally, placing the burden of compute and storage onto the cloud means that the actual endpoint devices can be less powerful. Less power translates into cheaper and more power-efficient devices.

As the term implies, the IoT is about the Internet *and* the things connected to the Internet. Accordingly, this will mean having code running on the device and services in the cloud. The ultimate goal of this project is to have multiple devices with Wi-Fi adapters scanning continuously for wireless networks. These devices will report their findings in real time. For the purposes of this article, we will assume that your device has access to the Internet.

The solution is to use a Microsoft Azure IoT Hub and Azure Stream Analytics. An Azure IoT Hub is a service that's highly scalable and can ingest millions of events per second. The IoT Hub is similar to the Azure Event Hub, except that it has extra functionality for device management. Devices have to be registered in order to connect to the

Code download available at bit.ly/1UrY3rz.

IoT Hub. Once registered, a device transmits a message to the IoT Hub, which then collects the data in real time. That data can be sent off to Azure Stream Analytics.

Azure Stream Analytics is a fully managed service that provides low-latency, high-availability, scalable, complex event processing over data streaming in from various sources. Stream Analytics will collect data streaming in from each one of the Wi-Fi scanner devices running the Wi-Fi Scanner UWP app.

Azure IoT Hub
collects the data from
registered devices
and Azure Stream
Analytics lets you
easily query the data.

In short, an Azure IoT Hub collects the data from registered devices and Azure Stream Analytics lets you easily query the data. To do this, you'll need to set up the appropriate services in Azure and make modifications to the UWP app.

Setting Up the Cloud Services

The first order of business is to create an Azure IoT Hub. To do this, browse to the Azure Portal, click New and then click Internet of Things to expand the options in that category, as shown in Figure 1. Click IoT Hub to see the dialog, as shown in Figure 2.

In this form, you'll enter a unique name for your IoT Hub. I chose CityWifiScannerHub. For now, I chose the free tier of service, because it's adequate for development. I also chose to create a new resource group and name it City-WifiScannerRG. It's best to pick a naming convention and stick with it. As for Region, it really doesn't matter which region you choose. Just make sure you pick the same region for all your services to save on data egress charges. For my purposes, East US will do nicely. Check the Pin to dashboard option for easier access to this service. Now, click on Create and the service will be up and running in a few moments.

Figure 2 The Create IoT Hub Dialog

While you wait for the service to start, now would be a good time to download the Device Explorer from bit.ly/25IXGY9.

Device Explorer

The Device Explorer is a utility that helps you build solutions with IoT Hub by letting you register and remove devices to your IoT Hub, as well as monitor data being sent between the device and the hub. To use the tool with the hub, you'll need to get the connection string from the IoT Hub. On the Azure Portal, browse to the IoT Hub and click the key icon. Because you want management privileges to add and remove devices, choose the iothubowner policy. In the following dialog, click the copy icon next to Connection string—primary key, as shown in Figure 3. This will copy the connection string to the clipboard.

With the connection string in the clipboard, go over to the Device Explorer and paste the connection string into the IoT Hub Connection String textbox. Click Update and you should see a message box stating that "Settings updated successfully." Click OK to close the message box.

Now, click on the Management tab. To register a device with this IoT Hub, you'll need to click Create. On the following dialog box, enter a Device ID, as you see in Figure 4. Click

the Create button to register the device with this service. A message box will appear to confirm that the operation was successful. Click Close to dismiss it.

You'll notice that there's an entry now in the Devices list. Right-click on the WiFiScanner1 entry. In the context menu that appears, choose Copy connection string for selected device, as shown in Figure 5. You'll use this connection string in the UWP app to connect it to the IoT Hub.

Figure 3 Getting the Connection String for Device Explorer

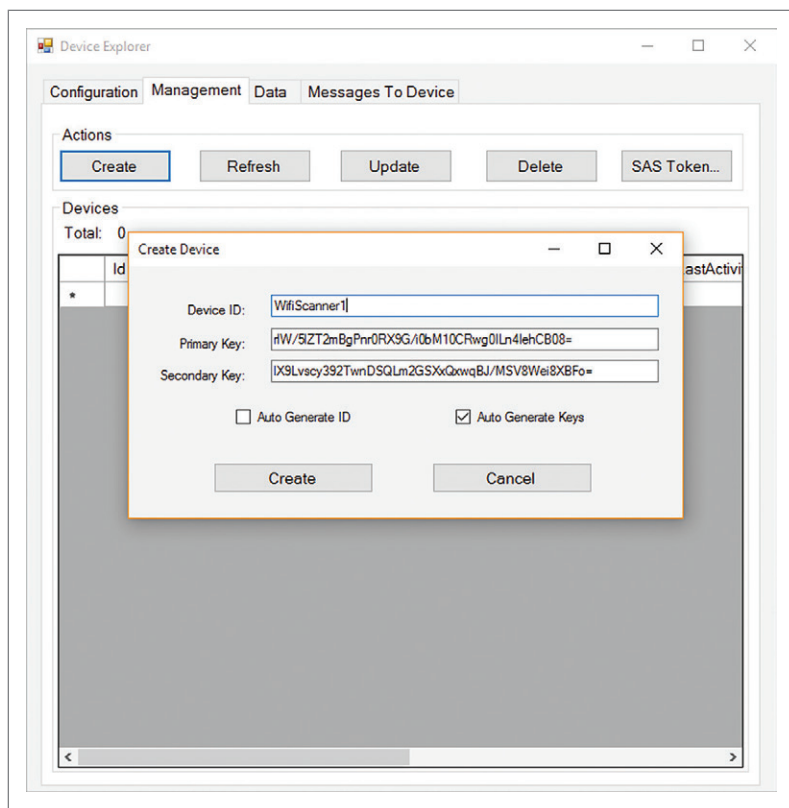


Figure 4 Registering a Device with Device Explorer

Modifying the UWP Code

This wouldn't be a column about UWP development without some UWP code. Now it's time to modify the app created in the previous column to connect it to the IoT Event Hub. Fortunately, Microsoft has created a library to make this simple and published it on NuGet. Load the app from the previous column into Visual Studio. Right-click on References, then click Manage NuGet Packages. In the search box, type Microsoft.Azure.Devices.Client and choose the first option. Install this package as you would any other NuGet Package.

Once the package is installed, add the following using statement to the MainPage.xaml.cs file:

```
using Microsoft.Azure.Devices.Client;
```

Then add the following member to the class, replacing "[ConnectionString]" with the connection string created in the previous section:

```
private DeviceClient deviceClient =  
    DeviceClient.CreateFromConnectionString(  
        [ConnectionString]);
```

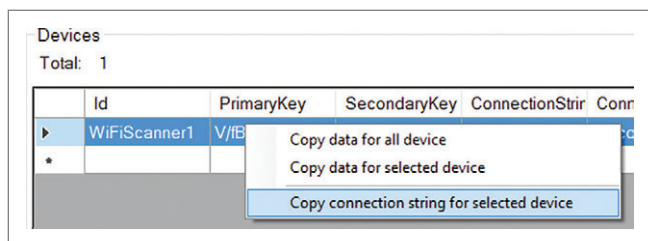


Figure 5 Getting the Connection String for the Device

The code needed to send the data up to the Azure IoT Hub is trivial thanks to the code inside the namespace Microsoft.Azure.Devices.Client. The following method does exactly that:

```
private async void SendScanData(string message)  
{  
    var content = new Message(  
        Encoding.UTF8.GetBytes(message));  
    await deviceClient.SendEventAsync(content);  
}
```

With this code in place in the MainPage class, you can now add the following lines toward the end of the RunWifiScan method of the MainPage.xaml.cs file. The first line in the following code converts the WifiPointData object to a JSON string for transport, while the second line calls the method that transmits the data up to the event hub:

```
string NetworkInfoJson = CreatedJson(wifiPoint);  
SendScanData(NetworkInfoJson);
```

Now it's time to run the app and send data up to the IoT Hub. But first, go to Device Explorer, click on the Data tab, and click the Monitor button. This will let you watch the data that gets sent to the IoT Hub.

Working with Stream Analytics

With all this work done, you have an IoT Hub in the cloud and a UWP app that sends data to it. Now what? The next step is to take the data and do something useful with it. For this task, you're going to use Stream Analytics. In the Azure Portal, click on New,

then search for Stream Analytics. Click Create and the New Stream Analytics Job dialog comes up. Name the job anything you like—I chose CityWifiScannerJob. I also chose to use the resource group created earlier. Once again, I chose East US to keep all my services for this in one datacenter. You should see something similar to **Figure 6**. Click Create to create the job.

The code needed to send the data up to the Azure IoT Hub is trivial thanks to the code inside the namespace Microsoft.Azure.Devices.Client.

Stream Analytics jobs work by taking data from an input source, processing it through a query, then sending it to an output. When you create a new job, you have zero inputs and outputs, along with a default query.

Create the Input

First, add an input by clicking on the Inputs tile and then on the Add button. In the New input dialog, give the input a name,



Get news from MSDN in your inbox!

Sign up to receive **MSDN FLASH**, which delivers the latest resources, SDKs, downloads, partner offers, security news, and updates on national and local developer events.

msdn
magazine

msdn.microsoft.com/flashnewsletter

as shown in **Figure 7** (I chose CitiWifiScannerInput). Change Source to IoT Hub. And then choose CityWifiScannerHub from the dropdown. Because the UWP app sends data in UTF-8-encoded JSON, there's no need to modify the default options. Click Create to create the input.

Stream analytics jobs work by taking data from an input source, processing it through a query and then sending it to an output.

Now that the input has been created, it's time to create the output. At press time, the new Azure Portal didn't support creating a Power BI output. For this, you'll need to load the classic portal from bit.ly/1V3IFPU. From there, go to the Stream Analytics section, choose the CityWifiScannerJob and click Output. In the list of options, choose Power BI. Name the output alias CitiWifiPowerBI, the Dataset Name CityWifiScans and the Table Name CityWifiTable. Click the next arrow. The following screen will ask you to authorize access to Power BI. You'll need to log into an organizational account. If you don't have one, follow the guidance in the blog post at bit.ly/29m89ZV to create an Office 365 trial account.

With the input and output in place, now it's time to edit the query. The full scope of what's possible with queries in Stream Analytics is beyond the scope of this article. Therefore, I'll create a very simple query that will send all the data from the input to the output without any transformations or clauses:

```
SELECT
*
INTO
CityWifiPowerBI
FROM
CitiWifiScannerInput
```

This syntax should look familiar to anyone who's used T-SQL. Note that the input and output names entered previously are

Figure 6 Configuring a New Stream Analytics Job

Figure 7 Creating the Input

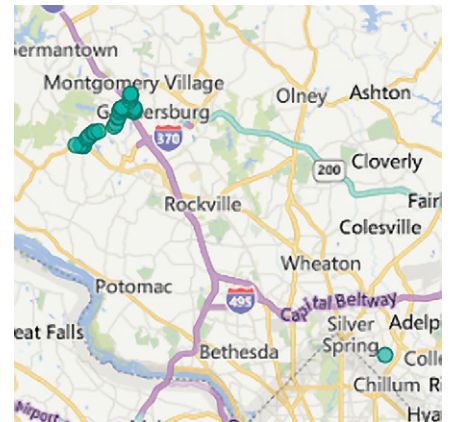


Figure 8 Creating a Map in Power BI with Data from the UWP App

used in the query to tell the job where to get the data and where to send it. Now, it's time to save the query and run the job. Click the Start button at the bottom of the page to start the job. It might take a few moments for the job to start. Once it does, open up Power BI in your browser. You should now see CitiWifiScans as a dataset in your workspace.

Open the dataset by clicking on it. Drag the Lat field from the Fields list onto the blank white space. A map control will appear. Now, drag the Long field onto the map control. You should see a map plotted with points of the latitude and longitude coordinates of your data. With my sample dataset, I have something that looks like **Figure 8**.

Wrapping Up

The real power and promise of the IoT revolution lies in the cloud. The cloud represents near limitless storage and processing power. That power and storage can be exploited by low-cost, power-efficient devices thanks to the Internet. From there, the data can be queued, processed and even visualized using tools like Power BI. Having the ability to easily modify a UWP app to send data to an Azure IoT Hub and deploy it to an IoT device makes for an unbeatable combination that will transform our world. ■

FRANK LA VIGNE is a technology evangelist on the Microsoft Technology and Civic Engagement team, where he helps users leverage technology in order to create a better community. He blogs regularly at FranksWorld.com and has a YouTube channel called Franks World TV (youtube.com/FranksWorldTV).

THANKS to the following technical experts for reviewing this article: Olivier Bloch and Rob Tiffany



ORLANDO
ROYAL PACIFIC RESORT AT
UNIVERSAL ORLANDO

**DEC
5-9**

*** REGISTER BY OCTOBER 5
AND SAVE \$400!**

SQL Server® **LIVE!**

TRAINING FOR DBAs AND IT PROS

Lead the Data Race

After 5 days of workshops, deep dives and breakout sessions, SQL Server Live! will leave you with the skills needed to Lead the Data Race.

With timely, relevant content, SQL Server Live! helps administrators, DBAs, and developers do more with their SQL Server investment. Sessions will cover performance tuning, security, reporting, data integration, adopting new techniques, improving old approaches, and modernizing the SQL Server infrastructure.



Use promo code SQLSEP1

Scan the QR code to register or for more event details.



A Part of Live! 360: The Ultimate Education Destination

6 GREAT CONFERENCES, 1 GREAT PRICE

Visual Studio **LIVE!**

ModernApps **LIVE!**

TECHMENTOR

Office & SharePoint **LIVE!**

SQL Server **LIVE!**

APPDEV TRENDS NEW!

SQLLIVE360.COM

EVENT PARTNERS



PLATINUM SPONSORS



GOLD SPONSORS



SILVER SPONSOR



SUPPORTED BY



PRODUCED BY





Mary Jane Grows Up

People of my generation remember the progression of marijuana, from a serious felony substance, to a misdemeanor, and now more or less legal in many places. I knew that legalization had reached its tipping point when I saw Microsoft entering the legal marijuana business.

“WHAT?!!” I hear you shout. “Microsoft? Weed?” Indeed. What the heck do you mean, Plattski?

As of this writing, the laws of 24 states, containing half the U.S. population, allow some form of legal marijuana, primarily medical. Of these, four states have legalized recreational use, including Microsoft’s home state of Washington. Two more states will be voting on medical marijuana this November, and four more on full legalization.

In states where weed is legal, the industry is tightly regulated. Weed can only be grown by certain licensed parties, sold by other licensed parties, taxes paid at various points; not unlike the booze industry. While couched in the language of public protection, most such regulation exists to ensure that the government gets its cut of the money—again, like booze. Weed itself may not be addictive, but the money it generates sure is.

As of this writing, the laws of 24 states, containing half the U.S. population, allow some form of legal marijuana, primarily medical.

It is this government regulatory apparatus that Microsoft now plans to serve (see nyti.ms/2a0BUmV). Microsoft will be working with a partner company named Kind (bit.ly/29EDs0D), which develops software that tracks marijuana plants from seed to sale. Microsoft will be packaging and marketing this software in its Azure Government cloud environment. (Insert obligatory clouds of smoke joke here.) Hmmm, Office365, Dynamics CRM and dope track, OK, whatever.

“We do think there will be significant growth,” said Kimberly Nelson, Microsoft’s executive director of state and local government solutions. “As the industry is regulated, there will be more transactions, and we believe there will be more sophisticated requirements and tools down the road.”

With flattening growth for its legacy products, opening this new channel into a rising industry makes good sense for Microsoft.

So Microsoft is helping state and local governments implement their laws. What’s the big deal? The answer is that marijuana is still banned under U.S. federal law, a Schedule I substance under the Controlled Substances Act. Even doctors can’t legally prescribe it. While the Obama administration announced seven years ago that it would not prosecute marijuana operations that comply with state laws, the conflicting regulations generate great uncertainty in the market. For example, many banks, leery of federal regulators combing their books, have refused service to this emerging industry, forcing state-legal marijuana businesses to operate solely with cash.

Microsoft is the first major company to step over this line and provide modern business services to this quasi-legal industry. But Microsoft’s impact goes far beyond the services themselves. Like diplomatic recognition of a new country, Microsoft’s public embrace projects a legitimacy that this infant industry desperately craves.

I’m guessing the federal laws likely will loosen, regardless of who wins the presidency in November. Surely a President Trump would pursue the money that legalized weed brings to state coffers. The feds currently don’t get a dime of it. And surely a President Clinton would want her husband to legally inhale next time.

As a father of teenagers, I find my feelings about legalized weed to be very different from those I harbored as a college student. I wouldn’t be happy to see my daughters smoking weed, but I’d be far less happy to see them thrown in the slammer for it. Marijuana exists (stop me if I get too technical here). I’d rather see it treated as a public health problem than a law enforcement problem.

I remember, right out of school, a friend of mine having a job offer rescinded because he admitted on his application to smoking marijuana. “Russ, it’s not the dope they care about,” I told him. “You just flunked the intelligence test by admitting it.” I’d rather see a job offer hinge on a programming test, wouldn’t you?

I remember Tech Ed in Atlanta in 2001. Microsoft’s keynote presentation featured a company that developed software for the booze industry, tying together the systems of the manufacturer, wholesaler and retailer. The audience marveled at their skill, and legally enjoyed some of the merchandise at that night’s reception. I wonder how long before we see this with weed? ■

DAVID S. PLATT teaches programming .NET at Harvard University Extension School and at companies all over the world. He’s the author of 11 programming books, including “Why Software Sucks” (Addison-Wesley Professional, 2006) and “Introducing Microsoft .NET” (Microsoft Press, 2002). Microsoft named him a Software Legend in 2002. He wonders whether he should tape down two of his daughter’s fingers so she learns how to count in octal. You can contact him at rollthunder.com.

Spreadsheets Made Easy.



Fastest Calculations

Evaluate complex Excel-based models and business rules with the fastest and most complete Excel-compatible calculation engine available.



Comprehensive Charting

Enable users to visualize data with comprehensive Excel-compatible charting which makes creating, modifying, rendering and interacting with complex charts easier than ever before.



Windows
Forms



Silverlight



WPF

Powerful Controls

Add powerful Excel-compatible viewing, editing, formatting, calculating, filtering, sorting, charting, printing and more to your WinForms, WPF and Silverlight applications.



Scalable Reporting

Easily create richly formatted Excel reports without Excel from any ASP.NET, Windows Forms, WPF or Silverlight application.

Download your free fully functional evaluation at SpreadsheetGear.com



SpreadsheetGear

Toll Free USA (888) 774-3273 | Phone (913) 390-4797 | sales@spreadsheetgear.com

SYNCFUSION SUCCINCTLY SERIES



97 titles and growing | Ad-free | 100 pages | PDF and Kindle formats

DOWNLOAD YOUR FREE COPY TODAY!

syncfusion.com/sdtGitHub



ASPOSE

File Format APIs

Powerful File APIs that are easy and intuitive to use
Native APIs for .NET, Java & Cloud

Using Aspose.Words for .NET to
Convert Word Docs to HTML -
Case Study

Adding File Conversion and
Manipulation to Business Systems

DOC, XLS, JPG,
PNG, PDF, BMP,
MSG, PPT, VSD,
XPS & many other
formats.



www.aspose.com

EU Sales: +44 141 628 8900

US Sales: +1 903 306 1676
sales@aspose.com

AU Sales: +61 2 8006 6987



Aspose.Total

Every Aspose API combined in one powerful suite.

➤ Aspose.Cells

XLS, CSV, PDF, SVG, HTML, PNG
BMP, XPS, JPG, SpreadsheetML...

➤ Aspose.BarCode

JPG, PNG, BMP, GIF, TIFF, WMF
ICON...

➤ Aspose.Words

DOC, RTF, PDF, HTML, PNG
ePub, XML, XPS, JPG...

➤ Aspose.Tasks

XML, MPP, SVG, PDF, TIFF
PNG...

➤ Aspose.Pdf

PDF, XML, XSL-FO, HTML, BMP
JPG, PNG, ePub...

➤ Aspose.Email

MSG, EML, PST, MHT, OST
OFT...

➤ Aspose.Slides

PPT, POT, ODP, XPS
HTML, PNG, PDF...


➤ Aspose.Imaging

PDF, BMP, JPG, GIF, TIFF
PNG...

and many more!



Contact Us:
US: +1 903 306 1676
EU: +44 141 628 8900
AU: +61 2 8006 6987
sales@aspose.com

 **ASPOSE**
File Format APIs



Working with Files?

Try Aspose File APIs

Convert
Print
Create
Combine
Modify



files from your applications!

Over 15,000 Happy Customers

.NET

Java

Cloud

Get your FREE evaluation copy at www.aspose.com

Aspose.Cells

Work with spreadsheets and data without depending on Microsoft Excel

- Solution for spreadsheet creation, manipulation and conversion.
- Import and export data.

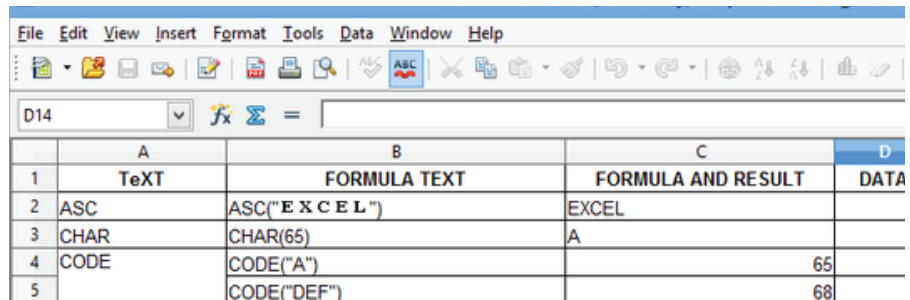
ASPOSE.CELLS IS A PROGRAMMING API that allows developers to create, manipulate and convert Microsoft Excel spreadsheet files from within their own applications. Its powerful features make it easy to convert worksheets and charts to graphics or save reports to PDF.

Aspose.Cells speeds up working with Microsoft Excel files. The

API is a flexible tool for simple tasks such as file conversion, as well as complex tasks like building models. Developers control page layout, formatting, charts and formulas. They can read and write spreadsheet files and save out to a wide variety of image and text file formats.

Fast, scalable, and reliable, Aspose.Cells saves time and effort compared to using Microsoft Office

A flexible API for simple and complex spreadsheet programming.



	A	B	C	D
1	TeXT	FORMULA TEXT	FORMULA AND RESULT	DATA
2	ASC	ASC("EXCEL")	EXCEL	
3	CHAR	CHAR(65)	A	
4	CODE	CODE("A")		65
5		CODE("DEF")		68

Aspose.Cells lets developers work with data sources, formatting, even formulas.

Automation.

Common Uses

- Building dynamic reports on the fly.
- Creating Excel dashboards with charts and pivot tables.
- Rendering and printing spreadsheets and graphics with high fidelity.
- Exporting data to, or importing from, Excel and other spreadsheets.
- Generating, manipulating and editing spreadsheets.
- Converting spreadsheets to images or other file formats.

Key Features

- A complete spreadsheet manipulation solution.
- Flexible data visualization and

reporting.

- Powerful formula engine.
- Complete formatting control.

Supported File Formats

XLS, XLSX, XLSM, XMPS, XLTX, XLTM, ODS, XPS, SpreadsheetML, tab delim., CSV, TXT, PDF, HTML, and many image formats including SVG, TIFF, JPEG, PNG and GIF.

Format support varies across platforms.

Platforms



Pricing Info					
	Standard	Enhanced		Standard	Enhanced
Developer Small Business	\$999	\$1498	Site Small Business	\$4995	\$7490
Developer OEM	\$2997	\$4494	Site OEM	\$13986	\$20972

The pricing info above is for .NET: prices for other platforms may differ. For the latest, contact sales.

www.aspose.com

EU: +44 141 628 8900

US: +1 903 306 1676
sales@aspose.com

Oceania: +61 2 8006 6987

Aspose.Cells for

.NET, Java, Cloud & more

File Formats

XLS, CSV, ODS, PDF, SVG, HTML, PNG, BMP, XPS, JPG SpreadsheetML and many others.

Spreadsheet Manipulation

Aspose.Cells lets you create, import, and export spreadsheets and also allows you to manipulate contents, cell formatting, and file protection.

Creating Charts

Aspose.Cells comes with complete support for charting and supports all standard chart types. Also, you can convert charts to images.

Graphics Capabilities

Easily convert worksheets to images as well as adding images to worksheets at runtime.

Get your FREE Trial at
<http://www.aspose.com>



100% Standalone

Aspose.Cells does not require Microsoft Office to be installed on the machine in order to work.

Aspose.Words

Program with word processing documents independently of Microsoft Word

- Solution for document creation, manipulation and conversion.
- Advanced mail merge functionality.

ASPOSE.WORDS IS AN ADVANCED PROGRAMMING API that lets developers perform a wide range of document processing tasks with their own applications. Aspose.Words makes it possible to generate, modify, convert, render and print documents without Microsoft Word. It provides sophisticated and flexible access to, and control over, Microsoft Word files.

Aspose.Words is powerful, user-friendly and feature rich. It saves developers time and effort compared to using Microsoft Office Automation and makes gives them powerful document management tools.

Aspose.Words makes creating, changing and converting DOC and other word processing file formats fast and easy.

Generate, modify, convert, render and print documents without Microsoft Word.

	Table				
	Column 1		Column 2	Column 3	Column 4
Row 1	Cell 1	Cell 2		Cell 3	Cell 4
Row 2	Cell 1		Cell 2	Cell 3	
Row 3	Cell 1		Cell 2		

Aspose.Words has sophisticated controls for formatting and managing tables and other content.

Common Uses

- Generating reports with complex mail merging; mail merging images.
- Populating tables and documents with data from a database.
- Inserting formatted text, paragraphs, tables and images into Microsoft Word documents.
- Adding barcodes to documents.
- Inserting diagrams and watermarks into Word documents.
- Formatting date and numeric fields.

Key Features

- A complete Microsoft Word document manipulation solution.
- Extensive mail merge features.
- Complete formatting control.
- High-fidelity conversion, rendering and printing.

Supported File Formats

DOC, DOCX, ODT, OOXML, XML, HTML, XHTML, MHTML, EPUB, PDF, XPS, RTF, and a number of image formats, including TIFF, JPEG, PNG and GIF.

Format support varies across platforms.

Platforms



Pricing Info					
	Standard	Enhanced		Standard	Enhanced
Developer Small Business	\$999	\$1498	Site Small Business	\$4995	\$7490
Developer OEM	\$2997	\$4494	Site OEM	\$13986	\$20972

The pricing info above is for .NET: prices for other platforms may differ. For the latest, contact sales.

www.aspose.com

EU: +44 141 628 8900

US: +1 903 306 1676
sales@aspose.com

Oceania: +61 2 8006 6987

Case Study: Aspose.Words for .NET

ProHire Staffing - Using Aspose.Words for .NET to convert Word Docs to HTML

PROHIRE IS THE WORKFORCE SOLUTIONS LEADER IN THE UNITED STATES AND SPECIALIZE IN THE RECRUITMENT OF SALES AND SALES MANGEMENT PROFESSIONALS.

We were founded with the goal of becoming the premier provider of executive search and placement services to the Fortune 500 and Inc. 500 Companies.

Problem

ProHire uses Bullhorn ATS as its Application Tracking System to track the electronic handling of its recruitment needs. We wanted to integrate the Bullhorn API with our new website. Our goal was to convert MS Word Documents resumes into a clean and concise HTML format into our existing .Net Stack. The converted HTML resume version needed to look close to the original.

Looking for a Solution

We chose the ASPOSE.Words product because it easily integrated into our existing .Net stack, and provided a quality MS Word to HTML conversion. The product was easy to download, and with a few lines of code we were up and running. We found the primary

difference between the Aspose.Words and other products was the obvious conversion quality from MS Word to HTML.

Finding a Solution

We had tested other products that converted Word to HTML. Every one we tested had some problem with the conversion. Some of them lost

elements of the resume during the conversion. Most of them changed the format of the resume or changed the color of the text unexpectedly. This is unacceptable when you are sending a resume to a hiring manger. We were very satisfied with the results. We did not need

any technical support because documentation was sufficient to us.

Implementation

Once we had the Aspose DLL our developer was able to implement Aspose.Words for .NET in a few hours. The transitions with Aspose.Words for .NET was very painless to do.

Outcome

We are very pleased with the success of our Aspose.Words for .NET implementation. Aspose.Words is a very powerful development tool that is well documented and easy to install. The documentation is easy to understand and use. If you want a product to convert Word Docs to HTML look no further. ProHire is happy to recommend Aspose.

This is an extract from a case study on our website. For the full version, go to: www.aspose.com/corporate/customers/case-studies.aspx

"The transitions with Aspose.Words for .NET was very painless to do."



The converted HTML resume version needed to look close to the original.

www.aspose.com



Open, Create, Convert, Print & Save Files

from within your *own* applications.

> ASPOSE.TOTAL

allows you to process these file formats:

- Word documents
- Excel spreadsheets
- PowerPoint presentations
- PDF documents
- Project documents
- Visio documents
- Outlook emails
- OneNote documents



**DOC XLS PPT PDF EML
PNG XML RTF HTML VSD
BMP & barcode images.**



ASPOSE

File Format APIs

Contact Us:

US: +1 903 306 1676

EU: +44 141 628 8900

AU: +61 2 8006 6987

sales@aspose.com

Helped over 11,000 companies and over 300,000 users work with documents in their applications.

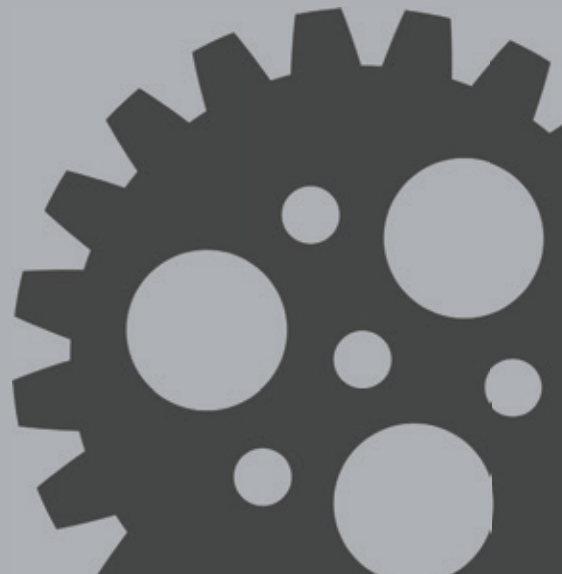


File Format APIs



GET STARTED NOW

- Free Trial
- 30 Day Temp License
- Free Support
- Community Forums
- Live Chat
- Blogs
- Examples
- Video Demos



Adding File Conversion and Manipulation to Business Systems

How often do people in your organization complain that they can't get information in the file format and layout they want? Converting documents from one format to another without losing layout and formatting should be simple, but it can be frustrating for both users and developers.

EXTRACTING DATA FROM A DATABASE AND DELIVERING IT TO THE SALES TEAM AS A REPORT, complete with charts and corporate branding, is fine. Until the sales team says that they want it as a Microsoft Excel file, and could you add a dashboard?

Using information from online forms in letters that can be printed and posted is easy. But what if you also want to add tracking barcodes and archive a digital copy as a PDF?

Ensuring that your business system supports all the different Microsoft Office file formats your users want can be difficult. Sometimes the native file format support of your system lets you down. When that is the case, use tools that extend that capability. A good tool can save you time and effort.

Document Conversion Options

Building your own solution: Time-consuming and costly, this option is only sensible if the solution you develop is central to your business.

Using Microsoft Office

Automation: Microsoft Office

Automation lets you use Microsoft Office programs server-side. It is not how the Office products were designed to be used. It can work well but you might notice issues with the stability, security and speed of the system, as well as cost.

Aspose creates APIs that work independently of Microsoft Office Automation.

Using an API: The API market has lots of free and commercial solutions, some very focused, some feature-rich. An API integrates with your code and gives you access to a range of new features.

Look to Aspose

Aspose are API experts. We create APIs, components and extensions that work independently of Microsoft Automation to extend a platform's native file format manipulation capabilities.

Aspose have developed APIs for .NET, Java, Cloud and Android that lets developers convert, create and manipulate Microsoft Office files – Microsoft Word, Excel, PowerPoint, Visio and Project – and other popular business formats, from PDFs and images to emails. We also have APIs for working with images,

barcodes and OCR. The APIs are optimised for stability, speed and ease of use. Our APIs save users weeks, sometimes months, of effort.



Finding the Right Tool

To find the product that's right for you, take a systematic approach:

- List must-have and nice-to-have features.
- Research the market.
- Ask for recommendations.
- Select a few candidates .
- Run trials.
- Evaluate
 - ease of use,
 - support and documentation,
 - performance, and
 - current and future needs.

www.aspose.com

EU: +44 141 628 8900

US: +1 903 306 1676
sales@aspose.com

Oceania: +61 2 8006 6987

Aspose.BarCode

A complete toolkit for barcode generation and recognition

- Generate barcodes with customer defined size and color.
- Recognize a large number of barcode types from images.

ASPOSE.BARCODE IS A ROBUST AND RELIABLE BARCODE GENERATION AND RECOGNITION API that allows developers to add barcode generation and recognition functionality to their applications quickly and easily.

Aspose.BarCode supports most established barcode specifications. It can export generated barcodes to multiple image formats, including BMP, GIF, JPED, PNG and TIFF.

Aspose.BarCode gives you full control over every aspect of the barcode

Robust and reliable barcode generation and recognition.

image, from background and bar color, through image quality, rotation angle, X-dimension, captions, and resolution.

Aspose.BarCode can read and recognize most common 1D and 2D barcodes from any image and at any angle. Filters help developers



Aspose.BarCode offers a large number of symbologies and formatting options.

clean up difficult to read images to improve recognition.

Common Uses

- Generating and recognizing barcode images.
- Printing barcode labels.
- Enhancing workflow by adding barcode functionality.
- Using recognition functions to drive real-life work processes.

Key Features

- Barcode generation and recognition.
- Comprehensive support for 1D and 2D symbologies.
- Image processing for improved recognition.

Supported File Formats

JPG, TIFF, PNG, BMP, GIF, EMF, WMF,

EXIP and ICON.

Format support varies across platforms.

Supported Barcodes

Linear: EAN13, EAN8, UPCA, UPCE, Interleaved2of5, Standard2of5, MSI, Code11, Codabar, EAN14(SCC14), SSCC18, ITF14, Matrix 2 of 5, PZN, Code128, Code39 Extended, Code39 Standard, OPC, Code93 Extended, Code93 Standard, IATA 2 of 5, GS1Code128, ISBN, ISMN, ISSN, ITF6, Pharmacode, DatabarOmniDirectional, VIN, DatabarTruncated, DatabarLimited, DatabarExpanded, PatchCode, Supplement **2D:** PDF417, MacroPDF417, DataMatrix, Aztec, QR, Italian Post 25, Code16K, GS1DataMatrix **Postal:** Postnet, Planet, USPS OneCode, Australia Post, Deutsche Post Identcode, AustralianPosteParcel, Deutsche Post Leticode, RM4SCC, SingaporePost, SwissPostParcel

Platforms



Pricing Info					
	Standard	Enhanced		Standard	Enhanced
Developer Small Business	\$599	\$1098	Site Small Business	\$2995	\$5490
Developer OEM	\$1797	\$3294	Site OEM	\$8386	\$15372

The pricing info above is for .NET: prices for other platforms may differ. For the latest, contact sales.

www.aspose.com

EU: +44 141 628 8900

US: +1 903 306 1676
sales@aspose.com

Oceania: +61 2 8006 6987









Aspose *for* Cloud

The easiest API to Create, Convert & Automate Documents in the cloud.



**Convert
Create
Render
Combine
Modify**

without installing anything!

Aspose.Words for Cloud  Create and convert docs Manipulate text Render documents Annotate	Aspose.Cells for Cloud  Create spreadsheets Convert spreadsheets Manipulate cells and formulas Render spreadsheets
Aspose.Slides for Cloud  Create presentations Manage slides Edit text and images Read and convert	Aspose.Pdf for Cloud  Create and convert PDFs Manipulate text, images Add pages, split, encrypt Manage stamps
Aspose.Email for Cloud  Create, update, and convert messages Extract attachments Use with any language	Aspose.BarCode for Cloud  Generate barcodes Read barcodes Set attributes Multiple image formats

Free Evaluation at www.aspose.com

Aspose.Email

Work with emails and calendars without Microsoft Outlook

- Complete email processing solution.
- Message file format support.

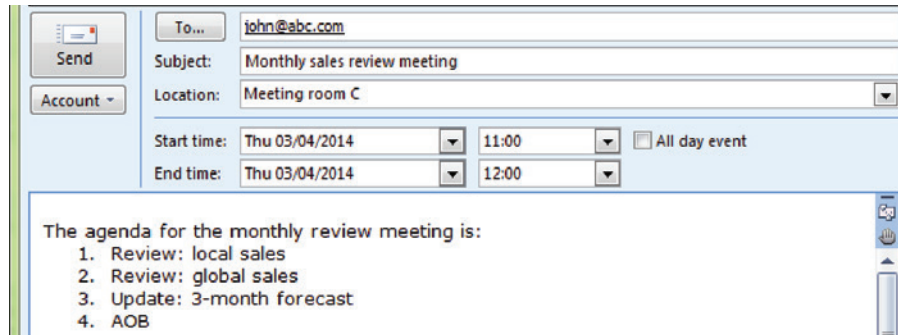
ASPOSE.EMAIL IS AN EMAIL PROGRAMMING API that allows developers to access and work with PST, EML, MSG and MHT files. It also offers an advanced API for interacting with enterprise mail systems like Exchange and Gmail.

Aspose.Email can work with HTML and plain text emails, attachments and embedded OLE objects.

It allows developers to work against SMTP, POP, FTP and Microsoft Exchange servers. It supports mail merge and iCalendar features, customized header and body, searching archives and has many other useful features.

Aspose.Email allows developers to focus on managing email without getting into the core of email and network programming. It gives you the controls you need.

Aspose.
Email works
with HTML
and plain
text emails,
attachments
and embedded
OLE objects.



Aspose.Email lets your applications work with emails, attachments, notes and calendars.

Common Uses

- Sending email with HTML formatting and attachments.
- Mail merging and sending mass mail.
- Connecting to POP3 and IMAP mail servers to list and download messages.
- Connecting to Microsoft Exchange Servers to list, download and send messages.
- Create and update tasks using iCalendar.
- Load from and save messages to file or stream (EML, MSG or MHT formats).

Key Features

- A complete email processing solution.
- Support for MSG and PST formats.
- Microsoft Exchange Server support.
- Complete recurrence pattern solution.

Supported File Formats

MSG, MHT, OST, PST, EMLX, TNEF, and EML.

Format support varies across platforms.

Platforms



Pricing Info					
	Standard	Enhanced		Standard	Enhanced
Developer Small Business	\$599	\$1098	Site Small Business	\$2995	\$5490
Developer OEM	\$1797	\$3294	Site OEM	\$8386	\$15372

The pricing info above is for .NET: prices for other platforms may differ. For the latest, contact sales.

www.aspose.com

EU: +44 141 628 8900

US: +1 903 306 1676
sales@aspose.com

Oceania: +61 2 8006 6987

Aspose.Pdf

Create PDF documents without using Adobe Acrobat

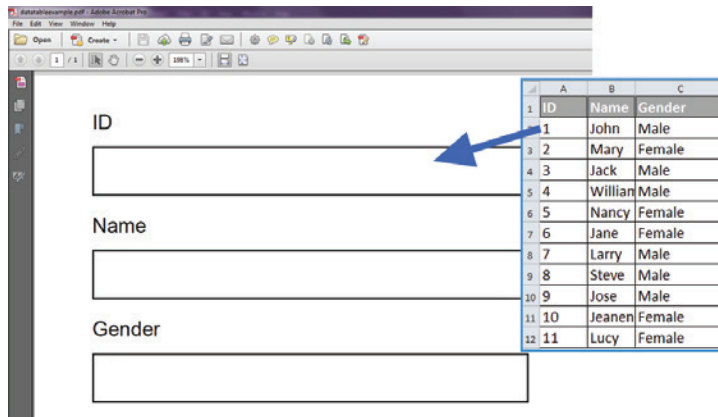
- A complete solution for programming with PDF files.
- Work with PDF forms and form fields.

ASPOSE.PDF IS A PDF DOCUMENT CREATION AND MANIPULATION API that developers use to read, write and manipulate PDF documents without using Adobe Acrobat. Aspose.Pdf is a sophisticated product that integrates with your application to add PDF capabilities.

Aspose.Pdf offers a wealth of features that lets developers compress files, create tables, work with links, add and remove security, handle custom fonts, integrate with external data sources, manage bookmarks, create table of contents, create forms and manage form fields.

Read, write and manipulate PDF documents independently of Adobe Acrobat.

It helps developers add, work with attachments, annotations and PDF form data, add, replace or remove text and images, split, concatenate,



Aspose.Pdf can be used to automatically complete PDF forms with external data.

extract or inset pages, and print PDF documents.

Common Uses

- Creating and editing PDF files.
- Inserting, extracting, appending, concatenating and splitting PDFs.
- Working with text, images, tables, images, headers, and footers.
- Applying security, passwords and signatures.
- Working with forms and form fields.

Key Features

- PDF creation from XML or XSL-FO documents.
- PDF form and field support.
- Advanced security and encryption.
- High-fidelity printing and conversion.
- Supported File Formats
- PDF, PDF/A, PDF/A_1b, PCL, XLS-FO, LaTeX, HTML, XPS, TXT and a range of image formats.

Format support varies across platforms.

Platforms



Pricing Info					
	Standard	Enhanced		Standard	Enhanced
Developer Small Business	\$799	\$1298	Site Small Business	\$3995	\$6490
Developer OEM	\$2397	\$3894	Site OEM	\$11186	\$18172

The pricing info above is for .NET: prices for other platforms may differ. For the latest, contact sales.

www.aspose.com

EU: +44 141 628 8900

US: +1 903 306 1676
sales@aspose.com

Oceania: +61 2 8006 6987

Aspose.Pdf

.Net, Java & Cloud

File Formats

PDF DOC XML XSL-FO XPS HTML BMP JPG PNG
ePUB & other image file formats.

Create and Manipulate PDFs

Create new or edit/manipualte existing PDFs.

Form Field Features

Add form fields to your PDFs. Import and export form fields data from select file formats.

Table Features

Add tables to your PDFs with formatting such as table border style, margin and padding info, column width and spanning options, and more.

Get started today at www.aspose.com



Aspose.Note for .NET

Aspose.Note for .NET is an API that lets developers convert Microsoft OneNote pages to a variety of file formats, and extract the text and document information.

Conversion is fast and high-fidelity. The output looks like the OneNote page, no matter how complex the formatting or layout.

Aspose.Note works independently of Office Automation and does not require Microsoft Office or OneNote to be installed.

Modify, convert, render and extract text and images from Microsoft OneNote files without relying on OneNote or other libraries.

Features

File Formats and Conversion

Microsoft OneNote
2010, 2010 SP1,
2013

Load,
Save

PDF

Save

Images (BMP, GIF,
JPG, PNG)

Save

Rendering and Printing

Save as Image
(BMP, GIF, JPG, PNG)

Save as PDF

Document Management

- Extract text
- Get the number of pages in a document.
- Get page information.
- Extract images.
- Get image information from a document.
- Replace text in document.

Aspose.Imaging

Create Images from scratch.

- Load existing images for editing purposes.
- Render to multiple file formats.

ASPOSE.IMAGING IS A CLASS

LIBRARY that facilitates the developer to create Image files from scratch or load existing ones for editing purpose. Also, Aspose.Imaging provides the means to save the created or edited Image to a variety of formats. All of the above mentioned can be achieved without the need of an Image Editor. It works independent of other applications and although Aspose.Imaging allows you to save to Adobe PhotoShop® format (PSD), you do not need PhotoShop installed on the machine.

Aspose.Imaging is flexible, stable and powerful. It's many features and image processing routines should meet most imaging requirements. Like all Aspose file format components, Aspose.

Imaging introduces support for an advanced set of drawing features along with the core functionality. Developers can

Create images from scratch. or load existing ones...

draw on Image surface either by manipulating the bitmap information or by using the advanced functionality like Graphics and Paths.

Common Uses

- Create images from scratch.
- Load and Edit existing images.
- Export images to a variety of formats.
- Adding watermark to images.
- Export CAD drawings to PDF & raster image formats.
- Crop, resize & RotateFlip images.
- Extract frames from multipage TIFF image.



Aspose.Imaging allows creation and manipulation of images.

Key Features

- Create, edit, and save images
- Multiple file formats
- Drawing features
- Export images

Supported File Formats

BMP, JPG, TIFF, GIF, PNG, PSD, DXF, DWG, and PDF.

Platforms



Pricing Info					
	Standard	Enhanced		Standard	Enhanced
Developer Small Business	\$399	\$898	Site Small Business	\$1995	\$4490
Developer OEM	\$1197	\$2694	Site OEM	\$5586	\$12572

The pricing info above is for .NET.

www.aspose.com

EU: +44 141 628 8900

US: +1 903 306 1676
sales@aspose.com

Oceania: +61 2 8006 6987

Aspose.Slides

Work with presentations without using Microsoft PowerPoint

- Complete solution for working with presentation files.
- Export presentations and slides to portable or image formats.

ASPOSE.SLIDES IS A FLEXIBLE PRESENTATION MANAGEMENT API that helps developers read, write and manipulate Microsoft PowerPoint documents. Slides and presentations can be saved to PDF, HTML and image file formats without Microsoft PowerPoint.

Aspose.Slides offers a number of advanced

features that make it easy to perform tasks such as rendering slides, exporting

Aspose.Slides gives you the tools you need to work with presentation files.

presentations, exporting slides to SVG and printing. Developers use Aspose.Slides to build customizable slide decks, add or remove standard graphics and automatically publish presentations to other formats.

Aspose.Slides gives developers the tools they need to work with presentation files. It integrates quickly and saves time and money.



Aspose.Slides has advanced features for working with every aspect of a presentation.

Common Uses

- Creating new slides and cloning existing slides from templates.
- Handling text and shape formatting.
- Applying and removing protection.
- Exporting presentations to images and PDF.
- Embedding Excel charts as OLE objects.
- Generate presentations from database.

Key Features

- A complete presentation development solution.
- Control over text, formatting and slide elements.
- OLE integration for embedding

external content.

- Wide support for input and output file formats.

Supported File Formats

PPT, HTML, POT, PPS, PPTX, POTX, PPSX, ODP, PresentationML, XPS, PDF and image formats including TIFF and JPG.

Format support varies across platforms.

Platforms



Pricing Info					
	Standard	Enhanced		Standard	Enhanced
Developer Small Business	\$799	\$1298	Site Small Business	\$3995	\$6490
Developer OEM	\$2397	\$3894	Site OEM	\$11186	\$18172

The pricing info above is for .NET: prices for other platforms may differ. For the latest, contact sales.

www.aspose.com

EU: +44 141 628 8900

US: +1 903 306 1676
sales@aspose.com

Oceania: +61 2 8006 6987

Support Services

Get the assistance you need, when you need it, from the people who know our products best.

- Free support for all, even when evaluating
- Get the level of support that suits you and your team

NO ONE KNOWS OUR PRODUCTS AS WELL AS WE DO.

We develop them, support them and use them. Our support is handled through our support forums and is available to all Aspose users.

Support

We are developers ourselves and understand how frustrating it is when a technical issue or a quirk in the software stops you from doing what you need to do. This is why we offer free support. Anyone who uses our product, whether they have bought them or are using an evaluation, deserves our full attention and respect. We have four levels of support that can fit your needs.

Everyone who uses Aspose products have access to our free support.



Work with the developers that developed and continue to maintain our products.

Support Options

Free

Everyone who uses Aspose products have access to our free support. Our software developers are on stand-by to help you succeed with your project, from the evaluation to roll-out of your solution.

Priority

If you want to know when you'll hear back from us on an issue and know that your issue is prioritized, Priority Support is for you. It provides a more formal support structure and has its own forum that is monitored by our software engineers.

Enterprise

Enterprise customers often have very specific needs. Our Enterprise Support option gives them access to the product development team and influence over the roadmap. Enterprise Support customers have their own, dedicated issue tracking system.

Sponsored

Available to Enterprise customers that would like to request features, this higher prioritized support can ensure your needed features are on our roadmap. A member of our team will produce a feature specification document to capture your requirements and how we intend to fulfill them so the direction development will take is clear up-front.



Pricing Info

To see the Priority and Enterprise support rates, refer to the product price list, or contact our sales team.

Sponsored Support is unique so pricing is specific to each project. Please contact our sales team to discuss.

www.aspose.com

EU: +44 141 628 8900

US: +1 903 306 1676
sales@aspose.com

Oceania: +61 2 8006 6987

We're Here to Help You

Aspose has 4 Support Services to best suit your needs

Free Support

Support Forums with no Charge

Priority Support

24 hour response time in the week,
issue escalation, dedicated forum

Enterprise Support

Communicate with product
managers, influence the roadmap

Sponsored Support

Get the feature you need built now

Technical Support is an issue that Aspose takes very seriously. Software must work quickly and dependably. When problems arise, developers need answers in a hurry. We ensure that our clients receive useful answers and solutions quickly.

Email • Live Chat • Forums

Contact Us

US Sales: +1 903 306 1676
sales@aspose.com

EU Sales: +44 141 628 8900

AU Sales: +61 2 8006 6987

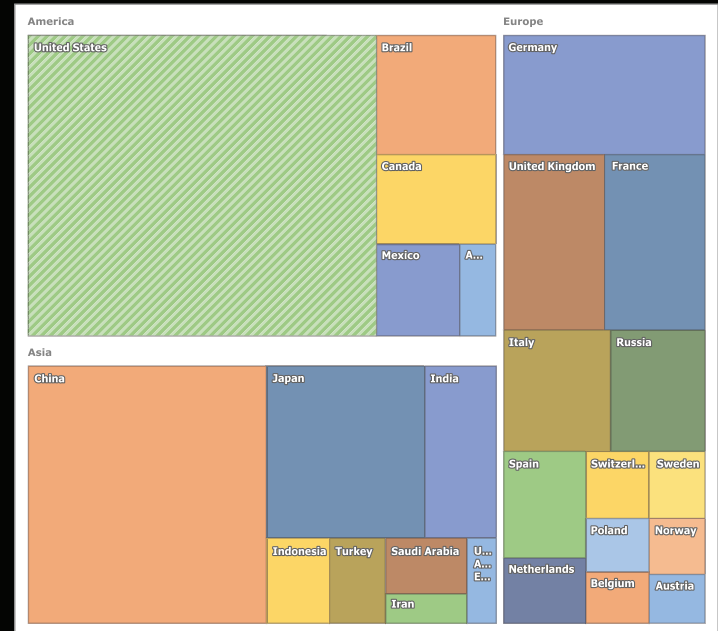


TreeMap for WinForms

NEW in v16.1 - The DevExpress TreeMap control provides the functionality to display your data in an easily readable visual form.



Part of the Best Selling **DevExpress Universal** Subscription



US Headquarters
ComponentSource
650 Claremore Prof Way
Suite 100
Woodstock
GA 30188-5188
USA

European Headquarters
ComponentSource
2 New Century Place
East Street
Reading, Berkshire
RG1 4ET
United Kingdom

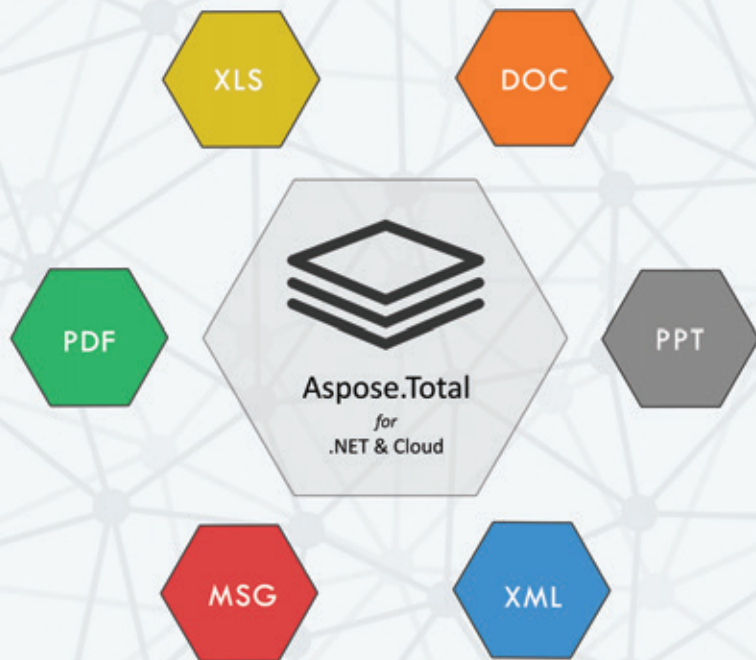
Asia / Pacific Headquarters
ComponentSource
7F Kojimachi Ichihara Bldg
1-1-8 Hirakawa-cho, Chiyoda-ku
Tokyo
Japan
102-0093

www.componentsource.com

Sales Hotline:
(888) 850-9911

Open Create Convert Print Save

files from your *applications!*



Trusted and used by 300,000+ Developers from 114 countries!

BUSINESS FILE FORMATS

Aspose.Cells

XLS, CSV, PDF, SVG, HTML, PNG
BMP, XPS, JPG, SpreadsheetML...

Aspose.Pdf

PDF, XML, XSL-FO, HTML, BMP, JPG, PNG, ePUB...

Aspose.BarCode

JPG, PNG, BMP, GIF, TIF, WMF, ICON...

Aspose.Email

MSG, EML, PST, MHT, OST, OFT...

Aspose.Words

DOC, RTF, PDF, HTML, PNG
ePUB, XML, XPS, JPG...

Aspose.Slides

PPT, POT, ODP, XPS, HTML, PNG, PDF...

Aspose.Tasks

XML, MPP, SVG, PDF, TIFF, PNG...

Aspose.Imaging

PDF, BMP, JPG, GIF, TIFF, PNG...

+ MANY MORE!

.NET










Cloud















Learn more at:
www.componentsource.com/aspose



ASPOSE
Your File Format APIs

What's in this Issue?

/n software	 /n software Red Carpet Subscription	6
Aspose	 Aspose.Words for .NET	8
	 Aspose.Pdf for .NET	10
CData Software	 SharePoint Drivers	12
	 Excel Services Drivers	13
	 Azure Table Drivers	14
	 Dynamics CRM Drivers	15
	 Dynamics GP Drivers	16
DevExpress	 DevExpress DXperience	18
	 DevExpress WPF	20
GrapeCity	 ComponentOne Ultimate	22
	 Spread Studio	24
Infragistics	 Infragistics WPF	26
	 Infragistics Ignite UI	28
	 Infragistics Ultimate	58
InstallAware	 InstallAware Studio	30
	 InstallAware Virtualization	32

Syncfusion	 Syncfusion Essential Studio Enterprise	34
	 Syncfusion Essential Studio for UWP	36
Text Control	 TX Text Control .NET for WPF Enterprise	38
	 TX Spell .NET for WinForms and ASP.NET	40
Actipro Software	 Actipro WPF Studio	42
Codejock Software	 Codejock Toolkit Pro	44
FusionCharts	 FusionCharts Suite XT	46
Janus Systems	 Janus GridEX for .NET	49
LEAD Technologies	 LEADTOOLS Document Imaging Suite SDK	50
Nevron	 Nevron Vision for .NET Ultimate	52
ComponentSpace	 ComponentSpace SAML Suite	54
Devart	 dbForge Developer Bundle for SQL Server	55
PDFlib	 PDFlib+PDI	56
JetBrains	 ReSharper Ultimate	57

© 1996-2016 ComponentSource. All Rights Reserved. All prices correct at the time of press. Online prices may vary from those shown due to daily fluctuations and online discounts.



Official Supplier

As official and authorized distributors, we supply you with legitimate licenses directly from 300+ software publishers.



24/5 Customer Service

Need help to find the right software license, upgrade or renewal? Call, Email or Live Chat with our experts.



Trusted for 20 Years

Over 580,000 licenses delivered to Developers, SysAdmins, Corporations, Governments & Resellers, worldwide.

Open 24 hours a day, Monday to Friday

International Customer Service Centers in the US, UK and Japan.

 United States 888 850 9911

 US Gov't Sales 888 850 9966

 United Kingdom 0800 581111

 Japan 0120 343 550

 Argentina 0800 666 0185

 Australia 1 800 0 15292

 Austria 0800 281 750

 Belgium 0800 7 5603

 Brazil 0800 891 6478

 Bulgaria 00800 115 4445

 Canada (888) 850 9911

 Chile 1230 020 6857


 China (North) 10800 481 1331

 China (South) 10800 813 1594

 Colombia 01800 710 2043

 Croatia 0800 22255

 Czech Republic 800 143 313

 Denmark 80 88 17 20

 Ecuador 1 800 010 562

 Finland 0800 1 18002

 France 0800 90 92 62

 Germany 0800 186 07 06

 Greece 00800 44121891

 Hong Kong 800 908 581


 Hungary 06800 16674

 India 000 800 44 01 455

 Ireland 1 800 535 661

 Israel 180 924 2003

 Italy 800 790046

 Japan 0120 343 550

 Korea, Rep. of 00798 14 800 6332

 Mexico 01 800 681 1559

 Netherlands 0800 022 8832

 New Zealand 800 449181

 Norway 800 1 3685

 Panama 00 1 800 203 1587


 Peru 0800 53288

 Poland 00800 442 1092

 Portugal 800 844 125

 Russia 810 800 2251 1044

 Singapore 800 810 2136

 Slovenia 0800 80297

 South Africa 0800 99 1088

 Spain 900 93 8926


 Sweden 020 794 989

 Switzerland 0800 83 5305

 Taiwan ROC 00 801 814313

 Turkey 00 800 4491 3617

 United Kingdom 0800 581111

 United States 888 850 9911

 Uruguay 000 401 902 38

 Venezuela 0 800 100 9103

US Headquarters

ComponentSource
650 Claremore Professional Way
Suite 100
Woodstock
GA 30188-5188
USA

Tel: +1 770 250 6100

Fax: +1 770 250 6199

Languages:

English / Spanish / Portuguese

Office Hours:

9:00am - 7:00pm EST

European Headquarters

ComponentSource
2 New Century Place
East Street
Reading
Berkshire
RG1 4ET
United Kingdom

Tel: +44 118 958 1111

Fax: +44 118 958 9999

Languages:

English / French / German /
Spanish / Italian

Office Hours:

9:00am - 5:30pm GMT & CET

Asia-Pac Headquarters

ComponentSource
7F Kojimachi Ichihara Bldg
1-1-8 Hirakawa-cho, Chiyoda-ku
Tokyo 102-0093
Japan

Tel: +81 3 3237 0281

Fax: +81 3 3237 0282

Languages:

English / Japanese / Korean /
Mandarin / Cantonese

Office Hours:

10:00am - 6:00pm JST

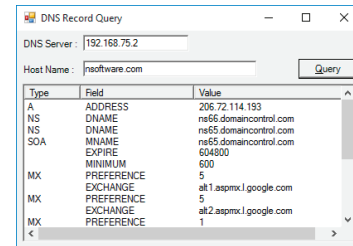
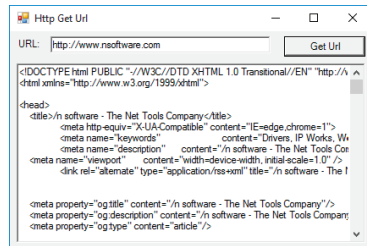
/n software Red Carpet Subscription

Comprehensive Internet components suite for any protocol, any platform, any IDE.

Publisher: /n software | Category: Communication & Messaging



Components for every major protocol from FTP to IMAP to SNMP, SSL and SSH security, S/MIME encryption, Digital Certificates, Credit Card Processing, Shipping and Tracking, and e-business (EDI) transactions.



Exceptionally Responsive

Based on a highly optimized exceptionally responsive asynchronous socket architecture, the components consume a minimum of system resources while providing a high degree of compatibility with existing systems.

Internet Protocol Components

Components for every major protocol from FTP to IMAP to SNMP, SSL and SSH security, S/MIME encryption, Digital Certificates, Credit Card Processing, ZIP compression and Shipping & Tracking.

Rigorously Tested

Rock solid components that have undergone thousands of hours of testing. Years of operation in Enterprise IT production environments add to the assurance that the components will perform as expected.



Prices from \$ 1,535.04
www.componentsource.com/nsoftware-red-carpet-subscription





We didn't invent the Internet...

...but our components help you
power the apps that bring it to business.

TOOLS • COMPONENTS • ENTERPRISE ADAPTERS

E-Business

AS2, EDI/X12, NAESB, OFTP ...

Credit Card Processing

Authorize.Net, TSYS, FDMS ...

Shipping & Tracking

FedEx, UPS, USPS ...

Accounting & Banking

QuickBooks, OFX ...

Internet Business

Amazon, eBay, PayPal ...

Internet Protocols

FTP, SMTP, IMAP, POP, WebDav ...

Secure Connectivity

SSH, SFTP, SSL, Certificatos ...

Secure Email

S/MIME, OpenPGP ...

Network Management

SNMP, MIB, LDAP, Monitoring ...

Compression & Encryption

Zip, Gzip, Jar, AES ...

Microsoft
NET

 SharePoint

 Windows Phone

 Microsoft
BizTalk Server

 Microsoft
SQL Server

 **activex**

connectivity
powered by 

To learn more please visit →

www.componentsource.com

Aspose.Words for .NET

Perform a wide range of document processing tasks within your .NET apps.



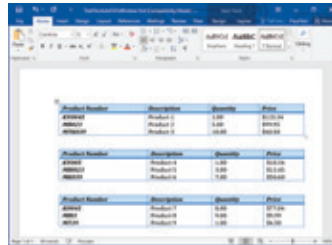
Publisher: Aspose | Category: Word Processing | ★★★★★

Create, edit, convert or print Word documents from any .NET framework based application. Aspose.Words for .NET supports a wide range of formats including DOC, DOCX, OOXML, RTF, HTML, OpenDocument, PDF, XPS and EPUB.



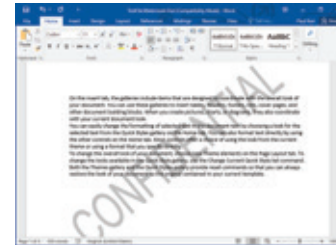
Reporting and Mail Merge

Design your reports in Microsoft Word and then allow Aspose.Words for .NET to populate documents with data from a number of data sources, including ADO.NET data tables and business objects.



Programming

Create, build and modify documents using an intuitive document object model that consists of 100+ classes. Programmatically create or modify all document elements including tables, text, fields and hyperlinks.



Render, Print & View Word Docs

Print or convert from Word documents to TIFF, PNG, BMP, JPEG, SVG or EMF images. You can also render any document page onto a .NET Graphic object and you can set its size and zoom level to create thumbnails.



Prices from \$ 979.02
www.componentsource.com/aspose-words-net



ASPOSE.TOTAL

Every Aspose component combined in ONE powerful suite!



Powerful File Format APIs

- ▶ **Aspose.Cells**
XLS, CSV, PDF, SVG, HTML, PNG
BMP, XPS, JPG, SpreadsheetML...
 - ▶ **Aspose.Pdf**
PDF, XML, XSL-FO, HTML, BMP
JPG, PNG, ePub...
 - ▶ **Aspose.Tasks**
XML, MPP, SVG, PDF, TIFF
PNG...
 - ▶ **Aspose.Words**
DOC, RTF, PDF, HTML, PNG
ePUB, XML, XPS, JPG...
 - ▶ **Aspose.Slides**
PPT, POT, ODP, XPS, HTML
PNG, PDF...
 - ▶ **Aspose.Email**
MSG, EML, PST, MHT, OST
OFT...
 - ▶ **Aspose.BarCode**
JPG, PNG, BMP, GIF, TIFF, WMF
ICON...
- ... and many others!

Aspose.Total for .NET

.NET

Aspose.Total for Cloud

Cloud

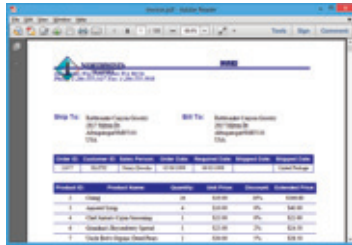
Aspose.Pdf for .NET

Create and manipulate PDF documents without using Adobe Acrobat.



Publisher: Aspose | Category: PDF | ★★★★★

Aspose.Pdf for .NET is a set of PDF APIs that enables your .NET applications to read, write and manipulate existing PDF documents. It also allows you to create forms and manage form fields embedded in a PDF document.



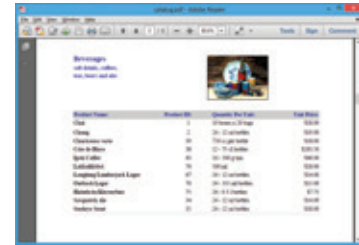
Convert PDFs

Convert between PDF and other popular document formats including HTML, PCL, DOCX, XLS, SVG, JPG, PNG, TIFF and XPS. You can also transform XML and XSL-FO documents into PDFs.



Table Creation

Access, create and modify tables, rows and cells. You can also add headings and Table of Contents during PDF creation and programmatically format document elements.



Document Manipulation

Concatenate or merge two or more PDF documents, append new pages to an existing PDF file and extract or insert pages. Document formatting is supported using an advanced document object model.



Prices from \$ 783.02
www.componentsource.com/aspose-pdf-net



Aspose.Cells for .NET, Cloud & more

File Formats

XLS, CSV, PDF, SVG, HTML, PNG, BMP, XPS, JPG, SpreadsheetML and many others.

Spreadsheet Manipulation

Aspose.Cells lets you create, import, and export spreadsheets and also allows you to manipulate contents, cell formatting, and file protection.

Graphics Capabilities

Easily convert worksheets to images as well as adding images to worksheets at runtime.

Creating Charts

Aspose.Cells comes with complete support for charting and supports all standard chart types. Also, you can convert charts to images.

No Office Automation

Aspose.Cells does not require Microsoft Office to be installed on the machine in order to work.

Learn more at:

www.componentsource.com/aspose



SharePoint Drivers



Connect to live SharePoint data through easy-to-use, bi-directional drivers.

Publisher: CData Software | Category: Data Access

Access SharePoint data as you would access a database. Comprehensive support for CRUD (Create, Read, Update, and Delete) operations. Integrate your applications with SharePoint Server Lists, Contacts, Calendar, Links, Tasks, and more.



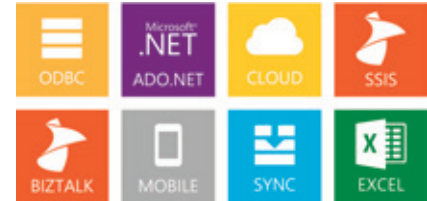
BI, Analytics & Reporting

Unlock the world of BI and Analytics, offering users live access to data from SharePoint through standard database drivers. Access SharePoint from Tableau, Qlik, Crystal Reports, Informatica, and many more.



Integrate with Custom Apps

The fastest and easiest way to integrate SharePoint connectivity from Web, Desktop, & Mobile Apps. Simply install the SharePoint drivers and access live data through tables, views and stored procedures.



Standards-based Drivers

Universally accessible drivers with bi-directional access to live applications, databases and Web APIs through familiar database interfaces, such as ODBC, ADO.NET, Cloud, SSIS, BizTalk, Mobile, Sync and Excel.



Prices from \$ 881.02
www.componentsource.com/cdata-sharepoint-drivers



Excel Services Drivers

Connect to SharePoint Excel Services through bi-directional data drivers.

Publisher: CData Software | Category: Data Access

Access live Excel Spreadsheet data through SharePoint Excel Services from BI, analytics, and reporting tools. Connect to SharePoint Excel Services the same way that you would connect to any other data source.



Standards-based Drivers

Universally accessible drivers with bi-directional access to live applications, databases and Web APIs through familiar database interfaces, such as ODBC, ADO.NET, SSIS, BizTalk and Mobile.



Integrate with Custom Apps

The fastest and easiest way to integrate Excel Services connectivity from Web, Desktop, & Mobile Apps. Simply install the Excel Services drivers and access live data through tables, views and stored procedures.



BI, Analytics & Reporting

Unlock the world of BI and Analytics, offering users live access to data from Excel Services through standard database drivers. Access Excel Services from Tableau, Qlik, Crystal Reports, Informatica, and more.



Prices from \$ 587.02
www.componentsource.com/cdata-excelservices-drivers



Azure Table Drivers



Connect to live Azure Storage data from BI, analytics, or reporting tools.

Publisher: CData Software | Category: Data Access

Azure Table Drivers hide the complexity of accessing Azure data, and provide additional powerful security features, smart caching, batching, socket management, and more.



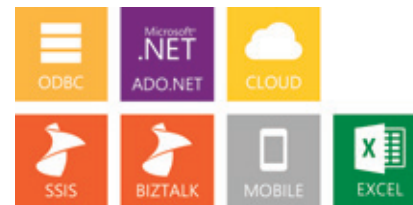
Integrate with Custom Apps

The fastest and easiest way to integrate Azure connectivity from Web, Desktop, & Mobile Apps. Simply install the Azure drivers and access live data through tables, views and stored procedures.



BI, Analytics & Reporting

Unlock the world of BI and Analytics, offering users live access to data from Azure through standard database drivers. Access Azure from Tableau, Qlik, Crystal Reports, Informatica, and many more.



Standards-based Drivers

Universally accessible drivers with bi-directional access to live applications, databases and Web APIs through familiar database interfaces, such as ODBC, ADO.NET, Cloud, SSIS, BizTalk, Mobile and Excel.



Prices from \$ 489.02
www.componentsource.com/cdata-azure-drivers



Dynamics CRM Drivers

Connect to Dynamics CRM data through easy-to-use, bi-directional drivers.

Publisher: CData Software | Category: Data Access

Dynamics CRM Drivers allows you to access Dynamics CRM data (OnPremise and Online) from reporting tools, databases, and custom applications.



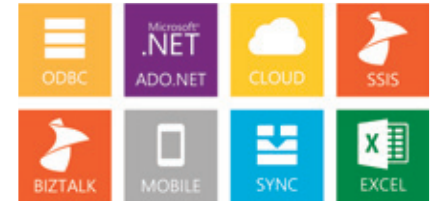
Integrate with Custom Apps

The fastest and easiest way to integrate Dynamics CRM connectivity from Web, Desktop, & Mobile Apps. Simply install the Dynamics CRM drivers and access live data through tables, views and stored procedures.



BI, Analytics & Reporting

Unlock the world of BI and Analytics, offering users live access to data from Dynamics CRM through standard database drivers. Access Dynamics CRM from Tableau, Qlik, Crystal Reports, Informatica, and many more.



Standards-based Drivers

Universally accessible drivers with bi-directional access to live applications, databases and Web APIs through familiar database interfaces, such as ODBC, ADO.NET, Cloud, SSIS, BizTalk, Mobile, Sync and Excel.



Prices from \$ 881.02
www.componentsource.com/cdata-dynamicscrm-drivers



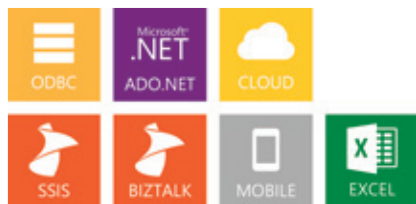
Dynamics GP Drivers



Access Microsoft Dynamics GP data from databases and custom apps.

Publisher: CData Software | Category: Data Access

Dynamics GP Drivers make integration a snap, providing an easy-to-use database-like interface to Microsoft Dynamics GP data including Vendors, Customers, Invoices, Quotes and more.



Standards-based Drivers

Universally accessible drivers with bi-directional access to live applications, databases and Web APIs through familiar database interfaces, such as ODBC, ADO.NET, Cloud, SSIS, BizTalk, Mobile and Excel.



BI, Analytics & Reporting

Unlock the world of BI and Analytics, offering users live access to data from Dynamics GP through standard database drivers. Access Dynamics GP from Tableau, Qlik, Crystal Reports, Informatica, and many more.



Integrate with Custom Apps

The fastest and easiest way to integrate Dynamics GP connectivity from Web, Desktop, & Mobile Apps. Simply install the Dynamics GP drivers and access live data through tables, views and stored procedures.



Prices from \$ 1,175.02
www.componentsource.com/cdata-dynamics-gp-drivers



SEE THE WORLD AS A DATABASE

ADO.NET • ODBC • SQL SSIS • ODATA
MYSQL • EXCEL • POWERSHELL



ODBC



ADO.NET



SSIS



CLOUD



BIZTALK



EXCEL



MOBILE

Work With Relational Data, Not Complex APIs or Services

Whether you are a developer building Apps with ADO.NET, OData, or Xamarin, or a systems integrator connecting systems with SQL SSIS or BizTalk, or even an information worker using ODBC or Excel — our state-of-the-art drivers give you live and direct access to

any data source. Get simple, standard SQL access to Microsoft Azure, Microsoft Dynamics, Microsoft Excel Services, Microsoft SharePoint, and much more.

Learn more: www.componentsource.com/cdata

Copyright © 2016 CData Software, Inc. All rights reserved. All trademarks and registered trademarks are the property of their respective owners.



SEE THE WORLD AS A DATABASE

DevExpress DXperience



All DevExpress .NET Framework control libraries in one integrated subscription.

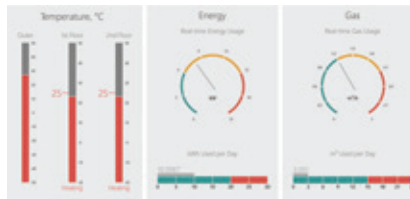
Publisher: DevExpress | Category: Presentation Layer | ★★★★★

DevExpress DXperience helps you build applications for Windows, Web, mobile and tablet with WinForms, ASP.NET, WPF and Windows 10 controls. Source code is included for all controls and libraries that ship as part of the Subscription.



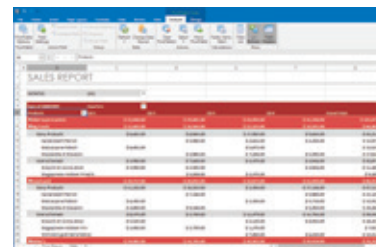
140+ WinForms Controls

A feature-complete and fully integrated development toolset designed to address the UI and Reporting needs of developers targeting the WinForms platform.



Windows 10 Apps

Create compelling touch-enabled business solutions that leverage the best parts of Windows 10 when targeting next generation devices such as Microsoft Surface.



90+ WPF Controls

Create highly functional XAML-MVM based applications to address any business need today and deliver next generation touch enabled solutions for tomorrow.

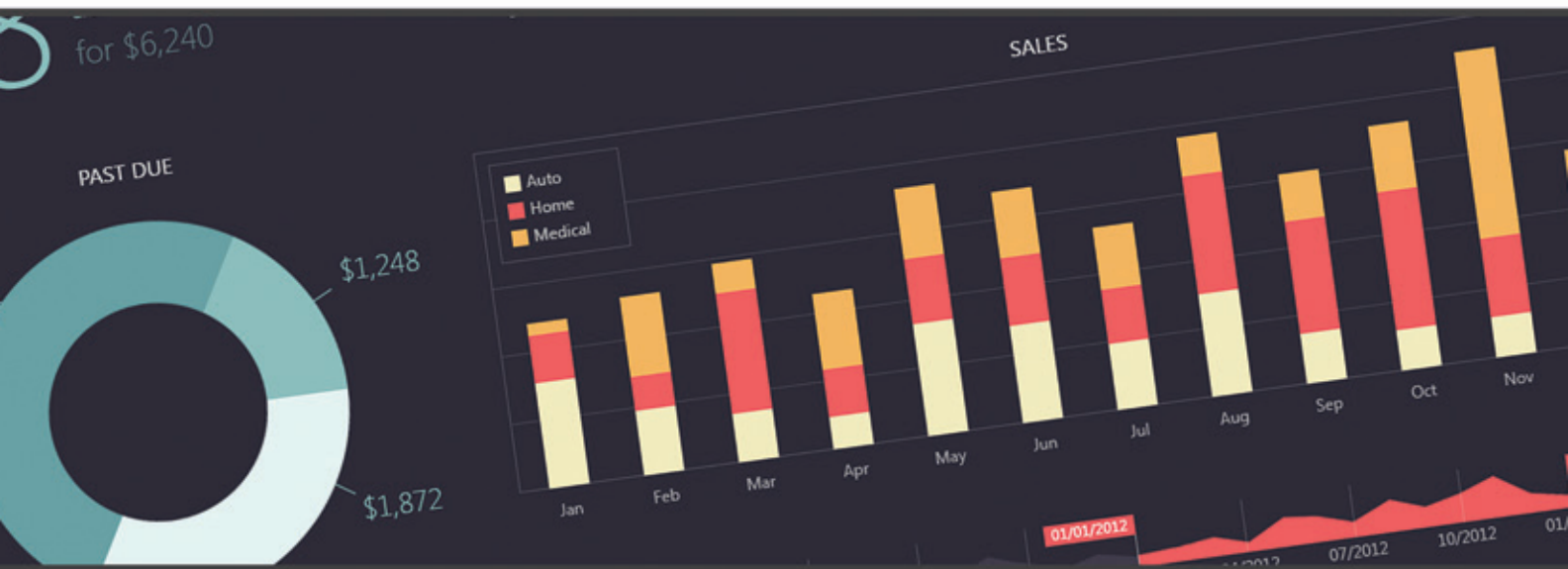


Prices from \$ 1,439.99
www.componentsource.com/devexpress-dxperience



Dashboards & Analytics

DevExpress Universal ships with everything you'll need to create information-rich decision support systems and to distribute your dashboard solutions royalty-free.



Your Next Great Dashboard Starts Here

Learn how you can create fully customizable Dashboards with our flexible data visualization tools.
componentsource.com/devexpress-universal



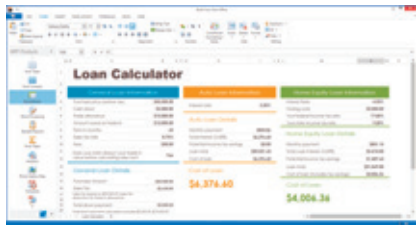
DevExpress WPF

Create compelling user-experiences on the WPF platform.



Publisher: DevExpress | Category: Presentation Layer

Whether you need to create an Outlook-inspired UI or need to touch-enable your app, DevExpress WPF will help you take your business solutions forward so you can deliver new, immersive experiences.



MVVM Inside

The Model-View-ViewModel (MVVM) pattern is the standard for building scalable, maintainable enterprise applications. DevExpress WPF includes comprehensive MVVM support to address any business challenge.



Touch-Enabled WPF Controls

DevExpress WPF Controls allow you to replicate the touch-first experience that lies at the heart of Windows 10 and to deliver highly responsive apps that meet the evolving needs of your enterprise.



Elegance is Built-in

DevExpress WPF ships with 15+ custom designed and highly polished themes for your next application. You can use each of these themes easily, without modification or manipulate them using the WPF Theme Editor.



Prices from \$ 863.99
www.componentsource.com/devexpress-wpf



Reporting Made Easy

Inform and analyze with absolute ease. DevExpress Reporting includes an enterprise-ready reporting platform with integrated Windows & Web report designers.

The screenshot displays a report designer interface. The main area shows an invoice template for 'NORTHWIND TRADERS'. The template includes a logo, company name, address, phone, fax, and email. It also features fields for 'Ship To' and 'Bill To' information, a 'Date' field, and a table with columns for 'Order ID', 'Customer ID', 'Salesperson', 'Order Date', 'Required Date', 'Shipped Date', 'Ship Via', 'Quantity', 'Unit Price', 'Discount', and 'Extended Price'. The 'Ship To' and 'Bill To' fields are populated with data from the 'Invoices' data source. The 'Date' field is set to '{0:d}'. The table is also populated with data from the 'Invoices' data source.

INVOICE

NORTHWIND TRADERS

One Porthall Way, Twin Points, WA 98156 Phone: 1-206-555-1417 Fax: 1-206-555-5938

Ship To:

- [Invoices.ShipName]
- [Invoices.ShipAddress]
- ShipCityRegionPostalCode
- [Invoices.ShipCountry]

Bill To:

- [Invoices.CustomerCompanyName]
- [Invoices.Address]
- CityRegionPostalCode
- [Invoices.Country]

Date: {0:d}

Order ID:	Customer ID:	Salesperson:	Order Date:	Required Date:	Shipped Date:	Ship Via:
[Invoices.OrderID]	[Invoices.CustomerID]	[Invoices.SalespersonID]	[Invoices.OrderDate]	[Invoices.RequiredDate]	[Invoices.ShipDate]	[Invoices.ShipperCode]

Quantity:	Unit Price:	Discount:	Extended Price:
[Invoices.Quantity]	[Invoices.UnitPrice]	[Invoices.Discount]	[Invoices.ExtendedPrice]

Properties Window:

- Text
- DATA BINDING
- SUMMARY
- Angle: 0
- Bookmark
- Parent Bookmark: Select...
- Auto Width: ☐
- Can Grow: ☒
- Can Shrink: ☐
- Multiline: ☒



Your Next Great Business Report Starts Here

See the power and flexibility of our reporting platform in action.

componentsource.com/devexpress-reporting



ComponentOne Ultimate



All ComponentOne Cross-Platform controls for Mobile, Web, and Desktop.

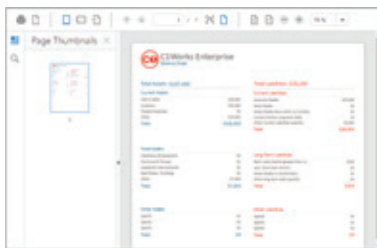
Publisher: GrapeCity | Category: Presentation Layer

Develop applications for any device, on any platform. Includes ComponentOne Studio - .NET controls for Visual Studio, Wijmo - next-generation HTML5/JavaScript controls, and Xuni - native mobile controls for iOS, Android, and Xamarin.



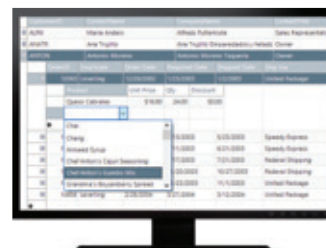
Cross-Platform Controls

Cross-platform, native mobile controls that employ the same API across iOS, Android, Xamarin. Includes fast, flexible, controls that deliver the same experience across all platforms.



HTML5/JavaScript Controls

A new generation of controls written in TypeScript for Enterprise Application development. With touch-first design, and full AngularJS support, the controls focus on top performance and zero dependencies.



.NET Controls

Build complete experiences in Visual Studio with fast, flexible .NET UI controls for mobile, web, and desktop. Add touch support to your desktop apps and keep your UI current with Microsoft UI & Office themes.



Prices from \$ 1,755.60

www.componentsource.com/componentone-ultimate



Get the Complete UI Control Toolkit

Deliver Awesome Windows, Web, and Mobile Apps

Newly Added FlexChart for UWP, WPF, and WinForms: This fast-flexible chart control lets you visualize data in your apps with a variety of customizable chart types.

Develop modern, streamlined, futureproof .NET applications with ComponentOne Studio's powerful, lightweight Microsoft Visual Studio controls. Deliver quality and high-performing desktop, web, and mobile apps.

Explore the flexibility of our modular references and API across all controls and platforms



Grids & Data Management

Get every spreadsheet function you need with our datagrid control - without any heavy lifting.



Data Visualization

Present large data sets out-of-the-box with our beautiful, flexible charts and gauges.



Reporting & Documentation

Migrate reports from another platform or generate your own with a full collection of controls.



Scheduling

Offer instant Outlook-like functionality in any application.

WinForms | WPF | UWP | ASP.NET MVC | ASP.NET Web Forms | LightSwitch | Silverlight | ActiveX

Learn more at: www.componentsource.com/componentone-studio

© 2016 GrapeCity. All rights reserved. All other product and brand names are trademarks and/or registered trademarks of their respective holders.



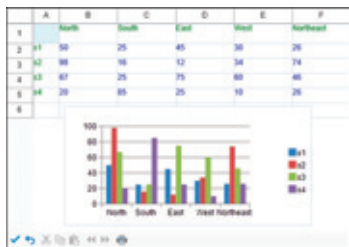
Spread Studio

.NET spreadsheet and grid components.



Publisher: GrapeCity | Category: Spreadsheet

Spread Studio is a multi-functional spreadsheet component that Visual Studio developers can use to create analysis, dashboard, data collection and management, scientific and financial applications for Windows and web.



Excel Compatibility

Import Excel files and keep formatting and security rules intact. Give your users the spreadsheet features they expect including drag-fill, drag move, hidden columns and rows, shapes and cell protection.

	A	B	C	D	E	F
1	Customer ID	Name	City	Sales Contact	Phone	Email
2	51342	Gary Jones	New York	Mark	(824) - 522-745	garyj@office.com
3	63452	Jason Green	New York	Mark	(824) - 563-356	jasong@office.com
4	45543	Sarah West	New York	Jackie	(824) - 567-567	sarahw@office.com
5	37687	James Cook	Buffalo	Mark	(856) - 689-654	jamesc@office.com
6	71242	Alan White	Buffalo	Jackie	(856) - 633-124	alanw@office.com
7	65467	Peter Roberts	Buffalo	Jackie	(856) - 637-354	peter@office.com
8	86758	Alison King	Buffalo	Mark	(856) - 637-378	alison@office.com
9	42534	Mary Adams	Philadelphia	Mark	(812) - 342-284	marya@office.com
10	98864	Justin Jones	Philadelphia	Jackie	(812) - 562-335	juston@office.com

Familiar Spreadsheet UI

Present data in a grid, a single spreadsheet, or in a multiple sheet workbook. However you choose to display data, you get features such as a function library, grouping, filtering, and conditional formatting.

	A	B	C	D	E	F
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						

Channel

Channel	Real	Virtual	Network	NM(db)
WRDC-DT	28	28.1		
WLFL-DT	27	22.1		
WRAL-TV	48	5.1		
WRAZ	49	50.1		

Print and Preview

Spread Studio fully supports printing and includes a print and preview window you can display in your applications. You can also customize print behavior by coding against the print API.



Prices from \$ 1,371.02
www.componentsource.com/spread-studio-for-net



NEW VERSION!

The reporting platform for all of your business needs

Design, publish, view, print and export operational reports such as invoices, expense reports, tax and government forms, as well as strategic and analytical reports, such as sales performance, budgeting, and revenue analysis. ActiveReports gives you the power and flexibility you need to turn your data into informative pixel-perfect reports across the enterprise.

SEVERITY LEGEND: ● Stable ●

DATE	FINDING	SEVERITY
Apr. 22nd, 2014 02:59PM	2nd Degree AVB Type I - Sinus Bradycardia w/ Degree AVB/3.5 Second Pause	●
Apr. 22nd, 2014 03:00PM	Atrial Flutter (Variable) w/ PVC/3.5 Sec Pause / Sinus Beats	●

Fiscal Year 2014 Financial Highlights

Quantity Sold by Region



15+ years on the market



300,000+ developer community



Multi-platform report viewers



Trusted by a worldwide customer base



Easy licensing

Learn more at
componentsource.com/activereports

© 2016 GrapeCity, Inc. All rights reserved. All other product and brand names are trademarks and/or registered trademarks of their respective holders.

 **GrapeCity.**
ActiveReports

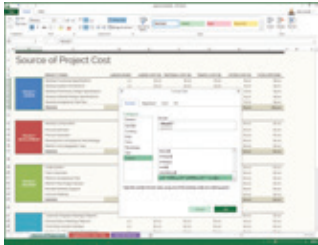
Infragistics WPF



Bring modern, engaging apps for desktop and touch screen devices to market.

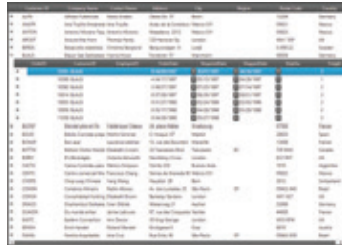
Publisher: Infragistics | Category: Presentation Layer

From fast data grids to dynamic data visualization, find the time saving controls you need with lightning-fast, touch-friendly WPF controls.



Familiar Interfaces

By embedding Microsoft Excel-like document create and editing experiences directly into your WPF applications you'll create products your users will love, complete with a full list of pixel-perfect features.



Speed and Scalability

When you have massive amounts of data that requires lightning-fast processing, only top-performing controls will do. That's where the WPF Data Grid comes in.



Touch Friendly

Take advantage of touch today. Infragistics WPF controls enable you to stay on trend and build touch-supported applications without having to add a single line of new code to your apps.



Prices from \$ 881.02
www.componentsource.com/infragistics-wpf





Build High Performance Desktop Apps



Give your users stylish, high-performance applications they'll love. Build dynamic apps for desktop and touch-screen devices with our lightning-fast, touch-friendly desktop controls. Cover every aspect of enterprise software development with the fastest data grids, Microsoft Office-inspired UI tools, potent data visualizations, and full modern experiences with touch & gestures.

Learn more about Infragistics WPF - part of Infragistics Ultimate
componentsource.com/infragistics

Infragistics Ignite UI

Create high-performance, touch-first, responsive apps.



Publisher: Infragistics | Category: Presentation Layer

A complete HTML & JavaScript toolkit to build modern browser experiences on any device - desktop, tablet or phone. Create high-performance, responsive apps with AngularJS directives, Bootstrap support and Microsoft MVC server-side widgets.



Lightweight and Fast

Ignite UI delivers best-in-class performance for your high-demand data needs. The virtualized, load-on-demand data grids can handle any data scenario.



Data Visualizations

With over 75 HTML5 charts, gauges, financial charts, statistical and technical indicators, trend lines and more, deliver the most demanding and beautiful touch-friendly dashboards for desktop and mobile apps.



Data Grids

Ignite UI includes a grid for flat data display, hierarchical master-detail, hierarchical tree-lists and even support for multi-dimensional OLAP data.



Prices from \$ 681.10
www.componentsource.com/infragistics-ignite-ui





Utilize the Power of jQuery/HTML5



Infragistics Ignite UI is the complete HTML & JavaScript toolkit to build modern browser experiences on any device – desktop, tablet or phone.

Learn more about Ignite UI - part of Infragistics Ultimate
componentsource.com/infragistics

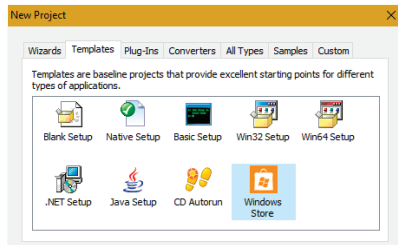
InstallAware Studio

A powerful software installation solution that supports the latest technologies.



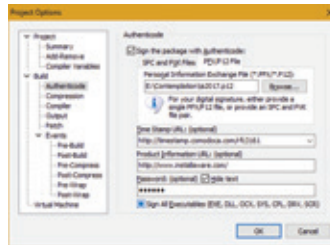
Publisher: InstallAware | Category: Release Automation & Management

InstallAware Studio is a software installation solution for Windows that provides you with the latest features and technology support, with the advantages of true rapid-Windows Installer development productivity.



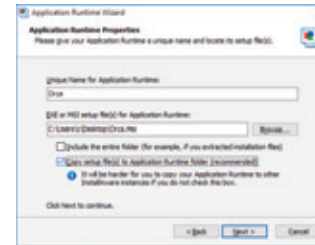
Windows Store Bridge

InstallAware effortlessly bridges your apps to the Windows Store, creating a Universal Windows app from a customizable template and helping your end-users download your apps directly from the Windows Store.



SHA 256 Code Signing

InstallAware applies a smart algorithm to determine whether files are going to be singly signed, doubly signed, or skipped. All signing happens automatically to comply with Microsoft's code signing standards.



Application Runtime Wizard

A point-and-click way to create new, reusable technology prerequisite installers for application frameworks in addition to those that ship built-in with InstallAware.



Prices from \$ 1,959.02
www.componentsource.com/installaware-studio





Only **InstallAware X5** Builds APPX,
App-V, MSI Windows Installer, EXE Native
Engine, and InstallAware Virtualization
Packages from a Single Source



InstallAware

- > Carry all your Win32, Win64, and .NET Apps into the Future!
- > Build any New or Existing InstallAware Setup as an APPX Package
- > Full Universal Windows Platform Support through Project Centennial
- > Package files, shortcuts, file type associations, and registry keys into APPX
- > Build on Windows 8.1 or newer, run on Windows 10 Redstone or newer

www.componentsource.com/installaware

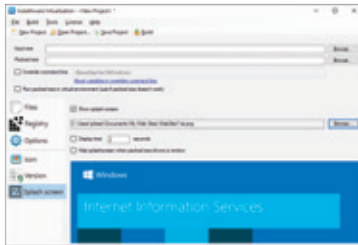
InstallAware Virtualization



Eliminate the need to install applications before running them.

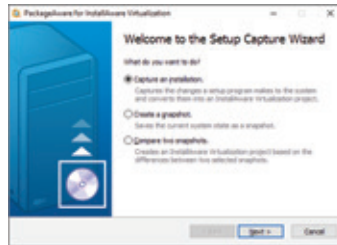
Publisher: InstallAware | Category: Release Automation & Management

Convert your full-fledged applications into single self-sustaining executable files that don't require installation or even pre-extraction to run. Squeeze all the resources that your application needs into a single executable.



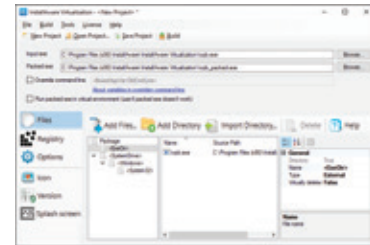
Eliminate Deployment Conflicts

InstallAware Virtualization allows you to run mutually incompatible applications on the same desktop at the same time. Completely sidestep all application deployment conflicts.



Application Repackager

InstallAware Virtualization eliminates the need to install applications before running them. Use PackageAware to easily capture any setup into a virtual application project.



Administrator Rights

Does your application require administrator rights? Virtualize it, and give it all the access it needs to the virtual file system and virtual registry; while keeping it away from protected areas of the host OS.



Prices from \$ 1,959.02
www.componentsource.com/installaware-virtualization



InstallAware X5 First With Complete Windows 10 Redstone Eco-System Support, Updated IDE with Snappier Designers; Touch and High-DPI Device Support

- › InstallAware X5's setup engine detects and installs on Windows 10 Redstone
- › Integrate with Visual Studio 15 and generate/build setups in a single click
- › Pre-Install the .NET 4.6.2 Framework, automatically detect dependencies
- › Pre-install Microsoft SQL Server 2016, 2014 SP2 with 50% better compression
- › InstallAware X5 IDE has a new Touch Mode and is faster than ever before

www.componentsource.com/installaware



Syncfusion Essential Studio Enterprise

Deliver better solutions with a complete suite of .NET & JavaScript controls.



Publisher: Syncfusion | Category: Presentation Layer | ★★★★★

Includes 650+ components across Windows Forms, WPF, ASP.NET MVC, ASP.NET Web Forms, LightSwitch, Silverlight, iOS, Android, Windows Phone, Xamarin, JavaScript and UWP. No royalties, run-time, or server deployment fees.



ASP.NET Web Forms

A comprehensive suite of ASP.NET Web Forms components for enterprise web development. It includes several complex widgets like data grid, chart, gantt, diagram, spreadsheet, schedule, pivot grid and much more.



Windows Forms

110+ enterprise class Windows forms components for developing modern desktop applications. Includes a powerful cell oriented grid control, Excel/Word/PDF/PowerPoint manipulation libraries a pivot grid and more.



Mobile Development

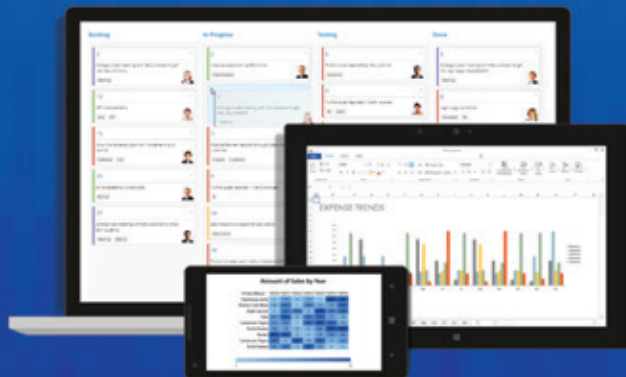
Essential Studio Enterprise comes with over 50 controls for developing Xamarin.Forms, Xamarin.iOS, and Xamarin.Android applications. It includes Excel, Word and PDF manipulation controls, data grids and charts.



Prices from \$ 1,795.50
www.componentsource.com/syncfusion-essential-studio-enterprise



MORE THAN **650** CUSTOMIZABLE CONTROLS AND FRAMEWORKS



ESSENTIAL STUDIO ENTERPRISE EDITION

WEB

ASP.NET MVC
ASP.NET Web Forms
ASP.NET Core
JavaScript
LightSwitch
Silverlight

MOBILE

JavaScript
Orubase
Universal Windows Platform
Windows Phone
WinRT
Xamarin

DESKTOP

Windows Forms
WPF
Universal Windows Platform

FILE FORMATS

Excel
PDF
Word
PowerPoint

DATA SCIENCE

Predictive Analytics

Includes free access to Syncfusion's

- Dashboard Platform
- Big Data Platform
- Report Platform

LEARN MORE:

componentsource.com/syncfusion



Syncfusion Essential Studio for UWP



A comprehensive suite of components for Universal Windows Platform.

Publisher: Syncfusion | Category: Presentation Layer

Syncfusion Essential Studio for UWP includes chart, grid, gauge, diagram, map, notification, navigation, menu and PDF viewer controls. All components render adaptively based on the current device family that they are being rendered on.



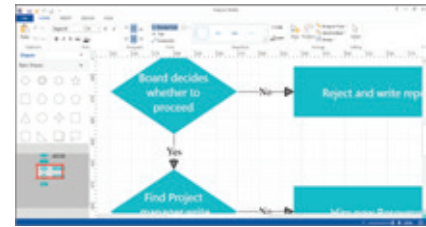
Chart

A high performance charting component that can plot 45+ chart types including specialized financial charts. Its rich feature set includes multiple axes, technical indicators, drilldown operations and zooming.

Order ID	Customer	Employee	Ship
10000	FRANK	EMP104	Ship
10001	FRANK	EMP106	Receiv
10002	FRANK	EMP108	Manag
10003	FRANK	EMP104	Ship
10004	WENEP	EMP112	Ship

Data Grid

The data grid control for UWP is used for displaying and manipulating tabular data. Its rich feature set includes data binding, editing, sorting, filtering, grouping, and exporting to Excel/PDF file formats.



Diagram

Create diagrams like flow charts, organizational charts, mind maps, floor plans, and swim lane diagrams either through code or through a visual interface that is similar to Visio.



Prices from \$ 975.10
www.componentsource.com/syncfusion-essential-studio-for-uwp



MORE THAN 50 CONTROLS

FOR XAMARIN.iOS, XAMARIN.ANDROID, AND XAMARIN.FORMS



ESSENTIAL STUDIO FOR XAMARIN

- The most feature-rich chart and data-grid components available.
- Create cross-platform mobile apps in C#.
- Design an attractive native UI for any device.
- Reuse your existing .NET code assets.

Also available as part of the
comprehensive Essential Studio
Enterprise Edition suite!

Get started today!

www.componentsource.com/syncfusion

 Syncfusion

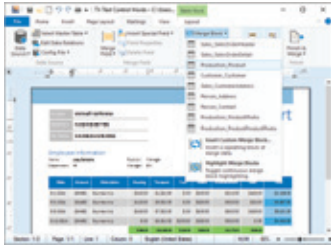
TX Text Control .NET for WPF Enterprise

Microsoft Word compatible WYSIWYG word processing.



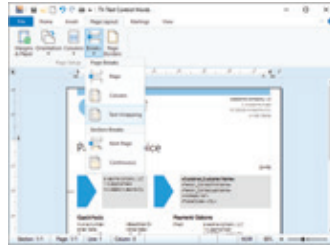
Publisher: Text Control | Category: Word Processing

TX Text Control .NET for WPF is a royalty-free, fully programmable rich edit control that offers developers a broad range of word processing features in a reusable component for Visual Studio.



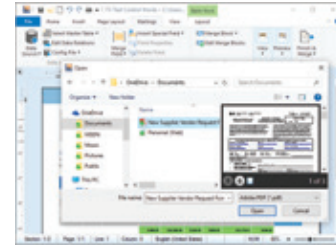
Reporting and Mail Merge

Create mail merge and reports, master-detail and sub-reports using MS Word compatible report templates. Data sources can be hierarchical such as related DataTables, DataSets or IEnumerable objects.



Page Section Support

Documents can be divided into an unlimited number of different sections, each one of which may be individually formatted. Sections enables users to create documents with mixed page orientation.



Import Adobe PDF Documents

Import "born digital" PDF documents, so that you can view, edit or convert these files. The PDF document is imported and can be modified just like any other format such as DOC or DOCX.

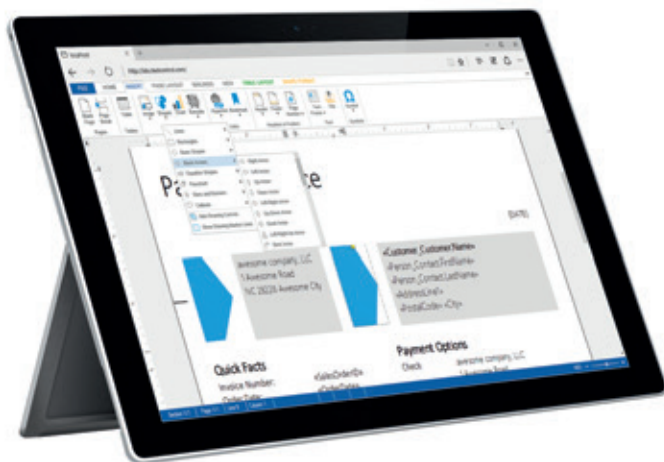
TEXTCONTROL

Prices from \$ 2,182.18

www.componentsource.com/tx-text-control-net-for-wpf-enterprise



ALL ABOUT DOCUMENTS

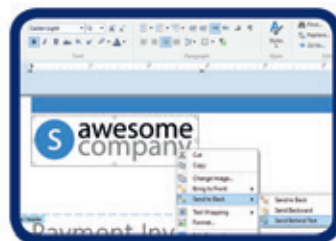
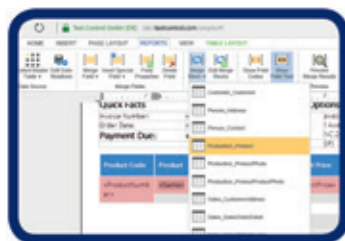


Text Control provides deep functionality components and consulting to integrate reporting and word processing into .NET Windows, web and mobile applications.

Reporting and Mail Merge

The Text Control Reporting Framework combines powerful reporting features with an easy-to-use, MS Word compatible word processor for ASP.NET, WPF and Windows Forms. Users can create documents and templates using ordinary Microsoft Word skills.

TX Text Control is completely independent from MS Word or any other third-party application and can be completely integrated into your business application.



MS Word compatible Rich Text Editing

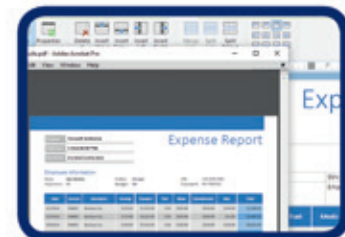
These feature-complete, fully programmable rich edit controls offer developers a broad range of word processing features in reusable UI and non-UI components for ASP.NET, WPF and Windows Forms.

They provide comprehensive text formatting, powerful mail merge features and all word processing key concepts such as table support, images, headers and footers, page sections and spell checking.

Load and Save Adobe PDFs

Using TX Text Control, Adobe PDF and PDF/A documents can be created including document access security settings and digital signatures.

PDF documents can be imported to be viewed, edited and converted. Edit PDF documents just like any other format such as DOC or DOCX. A fully featured API can be used to modify the content or to search through the document.



Find out more about TextControl:

www.componentsource.com/textcontrol



Software • Training • Consulting

TEXT CONTROL

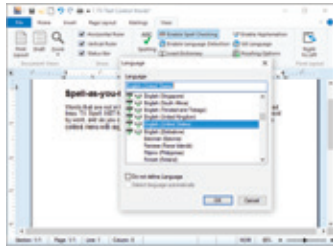
TX Spell .NET for WinForms and ASP.NET

Spell checking functionality for TX Text Control based applications.



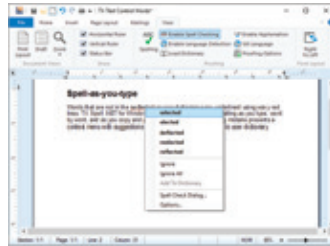
Publisher: Text Control | Category: Input & Data Validation

Add fast, reliable and accurate spell checking and language tools to your .NET based Windows Forms and ASP.NET applications. TX Spell .NET has been tested with over 60 freely available dictionaries.



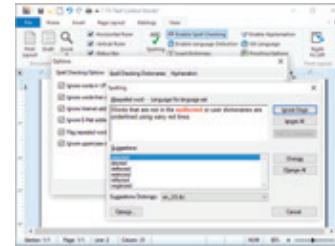
Multi-Lingual Spell Checking

TX Spell .NET supports multi-lingual documents. For example, it is possible to check a document, which contains English, German, French and Spanish text without having to manually switch between dictionaries.



Spell-as-you-type

Check spelling as you type, word by word, and as you copy and paste. Right-clicking on a spelling mistake presents a context menu with suggestions and options to Ignore All or Add to user dictionary.



Integrated Dialog Boxes

Using the integrated dialog box, you can see which words are incorrectly spelled, replace the word with one of the suggestions, ignore the spelling mistake just this once, or ignore it completely.

TEXTCONTROL

Prices from \$ 453.18
www.componentsource.com/tx-spell-net-for-windows-forms



ASP.NET MVC REPORTING

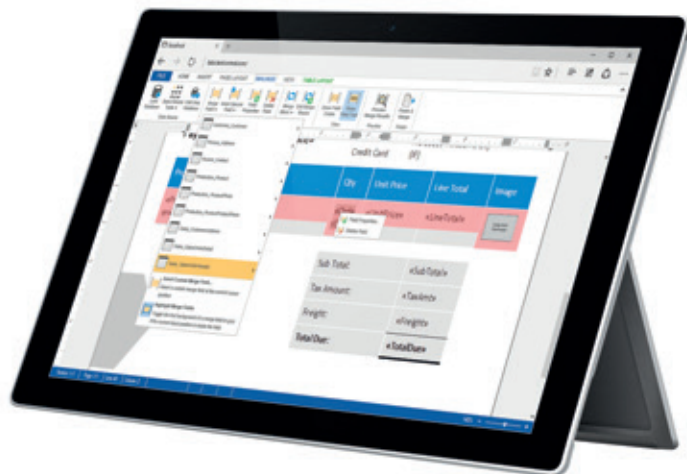
The first cross-browser, true WYSIWYG HTML5-based document editor.

Now with full ASP.NET MVC support.



Give your users an MS Word compatible editor to create powerful documents and reporting templates anywhere - in any browser. Feature-complete including mail merge, sections, headers and footers, drawings and shapes, tables, barcodes and charts.

Available for ASP.NET Web Forms and MVC.



Edit and create MS Word documents



Create and modify Adobe® PDF documents



Create reports and mail merge templates



Integrate with Microsoft® Visual Studio

PM> Install-Package TXTextControl.web

Find out more about TextControl:

www.componentsource.com/textcontrol



Software • Training • Consulting

TEXT CONTROL

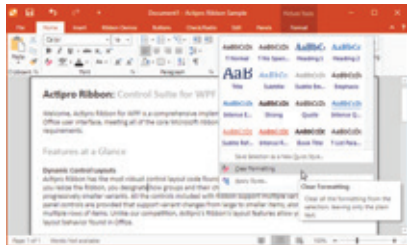
Actipro WPF Studio

Everything you need to add rich functionality to your WPF apps.



Publisher: Actipro Software | Category: Presentation Layer | ★★★★★

Over 100 WPF controls and components including barcodes, charts, datagrid, docking & MDI, editors, gauges, micro charts, navigation, property grid, ribbon, syntax editor, themes, views, wizards, and a shared library of controls.



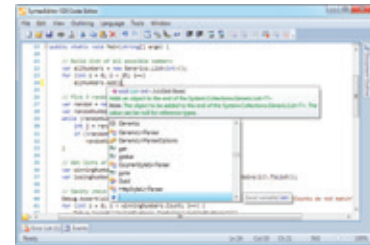
Ribbon Control

Use the Ribbon control to reproduce an Office-like user interface with split buttons, galleries and mini-toolbars. Various layout controls govern where items are placed within a ribbon as it decreases in width.



Visualize Complex Data

Actipro Charts is a set of full-sized charts that provide rich visualization for quantitative data. It's designed to make even the most complex data easily readable.



Code Editing in Your Apps

SyntaxEditor is a powerful text editing control that is packed with features for efficient code editing, including syntax highlighting, code outlining, parsing, line numbers and block selection.



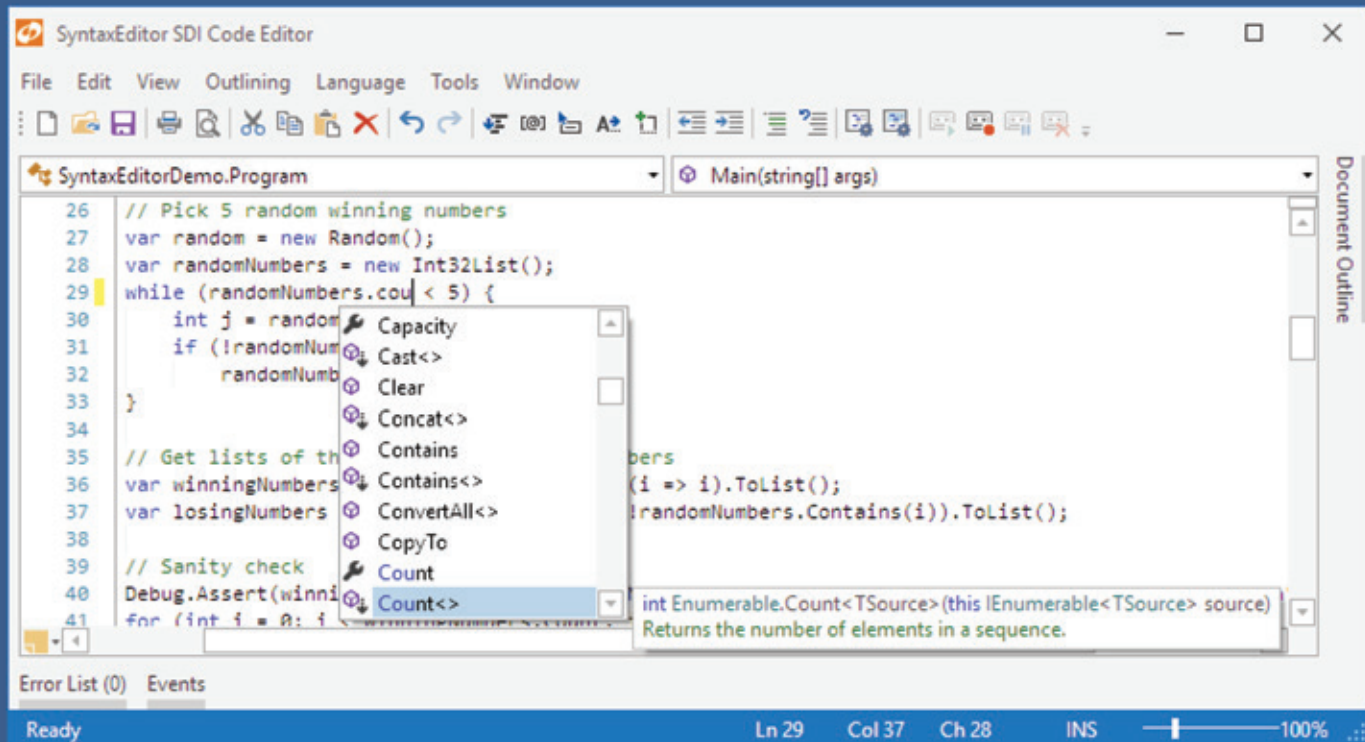
Prices from \$ 636.02
www.componentsource.com/actipro-wpf-studio



Actipro SyntaxEditor

for WPF, Universal Windows, Silverlight, WinForms

*The ultimate syntax-highlighting
code editor control*



Discover the possibilities...
componentsource.com/actipro



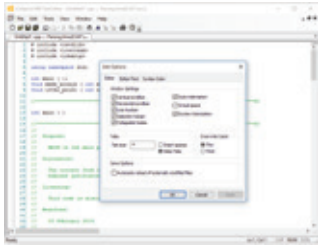
Codejock Toolkit Pro

Provides Visual C++ MFC developers with a professional set of modern controls.



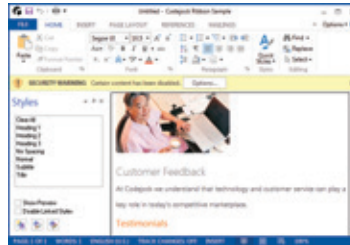
Publisher: Codejock Software | Category: Presentation Layer | ★★★★★

A robust set of components including: Command Bars, Controls, Chart Pro, Calendar, Docking Pane, Property Grid, Report Control, Shortcut Bar, Syntax Editor, Skin Framework and Task Panel.



Syntax Edit

Provides your users with a highly sophisticated text editor control that supports text block grouping, syntax colorization, line numbers, font type, pre-defined color schemes, book marks, break points and more.



Command Bars

Provides Windows developers with comprehensive, fully customizable menus, toolbars, ribbon and MDI tab windows to give your application a professional and modern interface.



Controls

Provides Windows developers with several ready-to-use components including Buttons, Combo Box, Common Dialogs, Progress Bars, Edit Controls, Alert PopUp Widows, Themed Scroll Bars, Tab Controls and more.

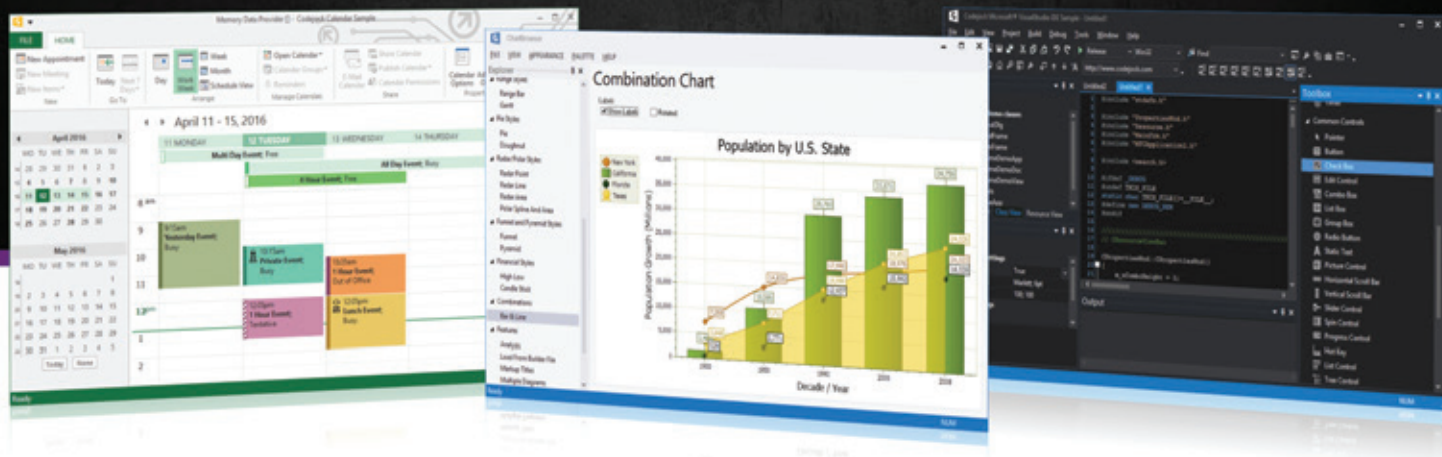


Prices from \$ 603.84
www.componentsource.com/codejock-toolkit-pro



Ribbons, Menus, Toolbars, Docking Panes, Calendars, Date Pickers, Grid Controls, Charts, Modern Themes and More!

for Visual C++ MFC / ActiveX COM



Toolkit Pro

for Visual C++ MFC

Suite Pro

for Active-X COM

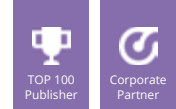
Create modern enterprise applications that incorporate a set of fully customizable user interface components including Microsoft® Office style ribbons, toolbars and menus. Many built in themes are available including Microsoft® Office and Visual Studio



Learn more at: componentsource.com/codejock



FusionCharts Suite XT



Create interactive JavaScript charts for Web, Mobile & Enterprise applications.

Publisher: FusionCharts | Category: Charts

FusionCharts Suite XT is a comprehensive JavaScript charting library, with over 90 charts and 1000 maps. With an extensive JavaScript API, it provides events, and methods for deep integration with other libraries and Web frameworks.



Works Everywhere

All charts work seamlessly across PCs, Macs, iPhones, iPads, and Android devices. All the charts are responsive to deliver the same experience everywhere.



Chart and Map Types

From basic charts (2D & 3D line, column, pie) to more complex ones (waterfall, gantt, candlestick, zoomline), FusionCharts Suite XT includes an exhaustive collection of JavaScript charts.



Chart Flexibility

FusionCharts gives you complete flexibility to customize the charts. You can centrally control the cosmetics of your charts like background color, plot colors, fonts etc. with the advanced theming engine.



Prices from \$ 587.02
www.componentsource.com/fusioncharts-suite





Delightful charts, gauges & maps in JavaScript / HTML5

FusionCharts Suite XT transforms all the boring data in your web and enterprise applications into delightful experiences. It is the perfect addition to all your reports, dashboards, monitors, analytics and surveys. FusionCharts Suite XT works seamlessly across PCs, Macs, iPads, iPhones and a majority of other mobile devices.

21,000 customers and 450,000 users in 118 countries trust and use FusionCharts Suite XT to add delight to their web and enterprise applications. The same delight that powers over a billion charts per month across the globe.

Start adding 'delight' to
your applications today!



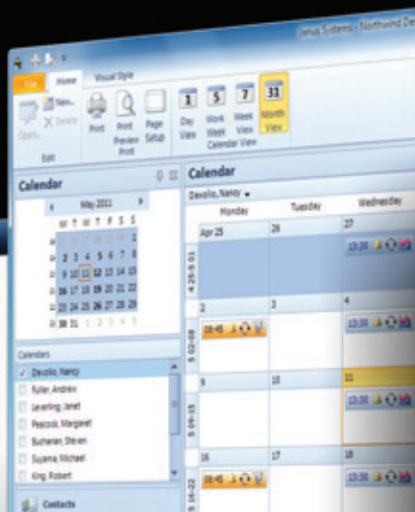
FusionCharts
Data to delight... in minutes

Janus WinForms Controls Suite v4

Add powerful Outlook style interfaces to your applications

Janus WinForms Controls Suite enhances your applications with a powerful user interface that features the familiar Outlook visual style. All controls are 100% .NET managed code, 64-bit compatible and support Visual Studio and .NET Framework Client Profiles.

Janus WinForms Controls Suite includes sample applications that let you quickly add Outlook look and feel to your applications.



Janus
systems

Learn more at:

www.componentsource.com/janus-systems

Janus GridEX for .NET

A data-aware, fully editable Outlook-style grid component.

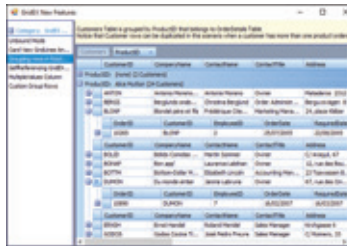
Publisher: Janus Systems | Category: Grids | ★★★★★

Janus GridEX implements its own data binding mechanism in order to perform sophisticated data manipulation. It also has an unbound mode for managing data in any format without sacrificing any group or sort capabilities.



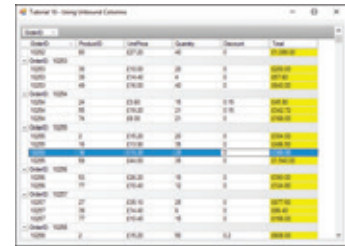
CardView

Janus GridEX for .NET CardView supports multicolumn layouts and the ability to expand or collapse cards. ColumnSets can also be used while the control is in CardView.



Filtering

Define simple or complex conditions that can be used to filter, format or find rows which conform to the criteria. The Control also presents a Filter Row which permits users to easily filter the data.



Hierarchical Data Display

Create effective Outlook-like user interfaces that display hierarchical data. The control allows you to display, edit, sort, group, filter, manipulate, summarize, preview and print your data.



LEADTOOLS Document Imaging Suite SDK

Develop powerful document imaging applications.



Publisher: LEAD Technologies | Category: Image Editing & Processing | ★★★★★

LEADTOOLS Document Imaging features PDF viewing and editing, comprehensive image annotation, specialized bitonal image display and image processing.



Document Viewer

Create robust, fully featured applications with rich-document viewing features, including text search, annotation, memory-efficient paging, inertial scrolling, and vector display.



Barcodes

LEADTOOLS supports more than 100 different barcode types and sub-types such as UPC, EAN, Code 128, QR Code, Data Matrix, and PDF417, making it the right choice for any barcode application in any industry.



Document Converter

Create powerful, automated document conversion, archival and delivery systems using .NET (C# & VB) for Windows, Android and Linux.



Prices from \$ 4,895.10
www.componentsource.com/leadtools-document-imaging-suite





Document Imaging

Document Viewer & Converter

OCR, MICR, OMR & ICR

1D & 2D Barcode

Forms Recognition

Create, Save, Edit & View PDF

Annotations & TWAIN



Medical Imaging

DICOM, CCOW & HL7

PACS Client & Server Framework

DICOMWeb (WADO)

Web & Desktop Viewers

Image Processing & Annotations

Medical 3D (MPR, MIP, VRT)



Multimedia Imaging

Play, Capture, Convert & DVR

MPEG2-TS & RTSP

Media Streaming Server

MPEG-4, H.264, H.265 & more

Media Foundation & DirectShow

Distributed Transcoding

.NET

Windows API

WinRT

Linux

iOS

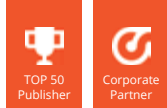
OS X

Android

JavaScript

Nevron Vision for .NET Ultimate

Data visualization components for Desktop and Web development.



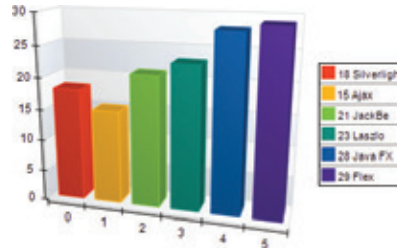
Publisher: Nevron | Category: Charts

Nevron Vision for .NET offers everything you need to create advanced digital dashboards, reports, diagrams and MMI interfaces. It includes charts, gauges, diagrams, maps and user interface controls.



Nevron Gauge for .NET

Suitable for any application that needs to visualize KPIs or Scorecards. It features a complete set of Radial and Linear gauges, LED displays, State Indicators, Advanced Axes and visual effects.



Nevron Chart for .NET

Display virtually any 2D, 3D Chart or Gauge with superior quality. Packed with hundreds of intuitive examples and many advanced features, Nevron Chart allows you to start quickly using no code at all.



Nevron Diagram for .NET

A complete diagramming solution packed with interactive features, shapes, automatic layouts, stunning visual effects and comes equipped with ready to use controls to boost your application development.

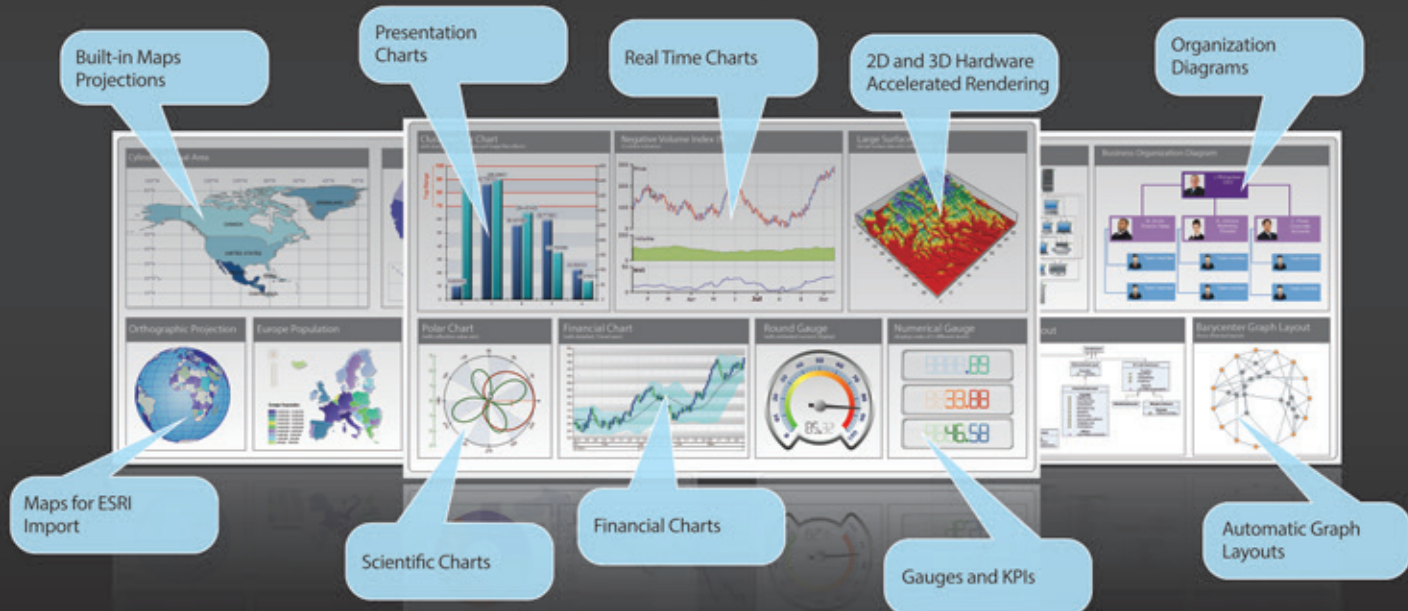


Prices from \$ 3,221.64
www.componentsource.com/nevron-net-vision-ultimate



Nevron Data Visualization

The leading data visualization components for desktop and web development in a single package.



solutions available for

Microsoft
ASP.net

Microsoft
Silverlight

WPF

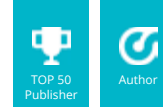
WinForms

SharePoint

Microsoft
SQLServer

Learn more at **componentsource.com/nevron** today

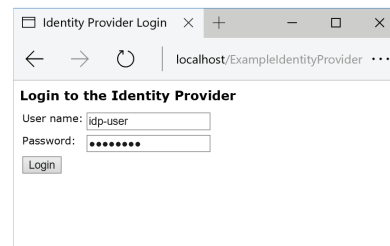
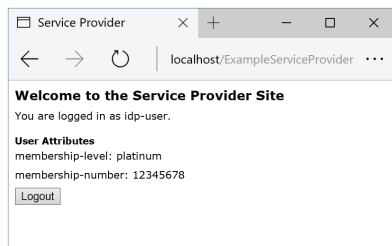
ComponentSpace SAML Suite



Implement SAML v2.0 and v1.1 assertions, protocol, bindings and profiles.

Publisher: ComponentSpace | Category: Communication & Messaging

ComponentSpace SAML Suite is fully compliant with the OASIS Security Assertion Markup Language (SAML) v2.0 and v1.1 specifications and offers .NET developers an easy way to manipulate SAML assertions from within their applications.



SAML Protocols

Create, modify and access SAML protocol messages and serialize SAML messages to and from XML. The easy to understand API requires only a few lines of code to SAML SSO enable your web application.

SAML SSO Solutions

Enables organizations to provide their customers, employees and partners with seamless, secure access to cloud and corporate web applications using a single username and password.

OASIS Compliant

SAML Suite is fully compliant with OASIS Security Assertion Markup Language specifications.

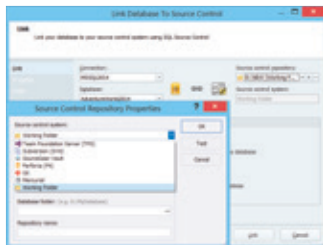


dbForge Developer Bundle for SQL Server

Version-control databases, compare data and optimize database performance.

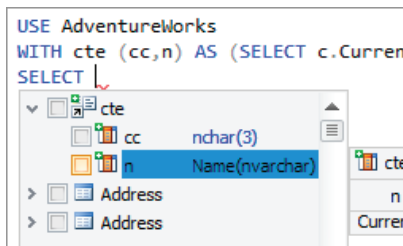
Publisher: Devart | Category: Database Development & Management

dbForge Developer Bundle for SQL Server includes several definitive add-ins that can turn your basic SQL Server Management Studio into a powerful platform allowing to cope with any database task.



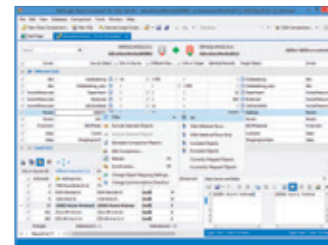
Source Control for SQL Server

A powerful SSMS add-in for managing SQL Server database changes. The tool can link your databases to all popular source control systems, and delivers smooth and clear workflow in a familiar interface.



dbForge SQL Complete

Provides a wide range of code completion features that relieve users from remembering long and complex object names, column names and SQL operators.



Data Compare for SQL Server

Tune your SQL database comparison, quickly analyze differences in a well-designed user interface and effortlessly synchronize data via a convenient wizard.



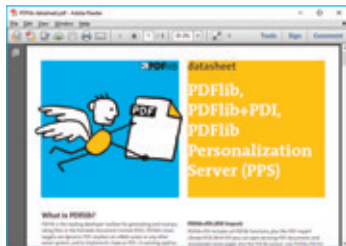
PDFlib+PDI

Developer toolbox for generating and manipulating PDF documents.

Publisher: PDFlib | Category: PDF



PDFlib lets you generate PDF documents with text, graphics, images, and interactive elements such as annotations or bookmarks. PDFlib+PDI adds the ability to add headers, footers, stamps, or page numbers to existing PDF pages.



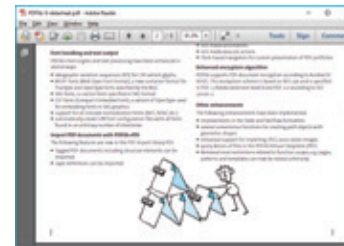
Generate PDFs

A component for generating and manipulating PDF files that frees you from the technicalities of the PDF file format. It leaves you to focus on acquiring data, arranging text, graphics and images on the page.



PDF/A Files

Create PDF/X-1/3/4/5 documents for commercial printing, PDF/A-1/2/3 documents for archiving, PDF/VT documents for transactional printing and Tagged PDF & PDF/UA documents for accessibility.



Convert to PDF

Convert TIFF, JPEG, or other image formats as well as SVG graphics to PDF. PDFlib easily integrates into all kinds of applications to enable reliable and high-quality PDF output.



Prices from \$ 2,177.78
www.componentsource.com/pdflibpdi



ReSharper Ultimate

Improve code quality, eliminate errors and analyze performance issues.



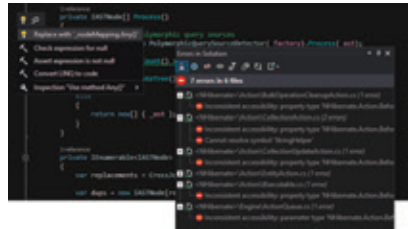
Publisher: JetBrains | Category: Coding & Debugging

ReSharper Ultimate combines all individual JetBrains .NET tools, as well as ReSharper C++. Each ReSharper Ultimate license allows a single developer to use ReSharper, ReSharper C++, dotCover, dotTrace, dotMemory and dotPeek.



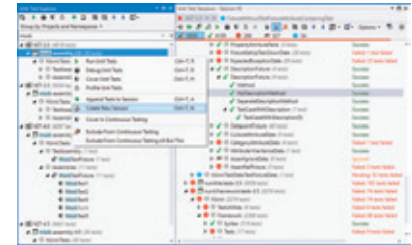
dotTrace

Detect performance bottlenecks in a variety of .NET applications: WPF and Universal Windows Platform applications, ASP.NET applications, Windows and WCF services, Silverlight and unit tests.



ReSharper

ReSharper extends Visual Studio with over 1,700 on-the-fly code inspections for C#, VB.NET, ASP.NET, JavaScript and TypeScript. For most inspections, ReSharper provides quick-fixes to improve your code.



Analyze .NET Code Coverage

Make sure you know to what extent your code is covered with dotCover. It integrates with Visual Studio so you can analyze and visualize code coverage without leaving the code editor.



Prices from \$ 391.02
www.componentsource.com/resharper-ultimate



Infragistics Ultimate



Accelerate your design and development for desktop, Web and mobile solutions.

Publisher: Infragistics | Category: Presentation Layer

Hundreds of UI controls including charts, maps, grids and themes. Supports ASP.NET & MVC, jQuery/JavaScript, WinForms, WPF, Xamarin.Forms, Android, and iOS. Also includes prototyping, data visualization, and mobile collaboration tools.



Design

Build interactive, responsive prototypes of your next Web, mobile or desktop application with Indigo Studio. Enjoy unlimited remote usability studies for user feedback on how your prototype will perform.



Develop

UX & UI tools built to accelerate the application development process for desktop, Web, mobile & hybrid apps. Ultimate now includes ReportPlus Embedded, allowing you to add powerful analytics to your next app.



Mobile Collaboration

Mobilize your SharePoint and Office 365 investment with SharePlus, a collaboration app for mobile devices. You can customize SharePlus using existing Web development skills to deliver native experiences.



Prices from \$ 1,955.10
www.componentsource.com/infragistics-ultimate





Great Apps Happen By Design



Design

Desktop

Web

Mobile

Data Visualization

Superior user experiences start with Infragistics Ultimate.
Choose UX & UI tools built to accelerate the application design and development process.

Learn more about Infragistics Ultimate at
componentsource.com/infragistics

Build User Experiences

UI Controls and Developer Productivity Tools for
Delivering Apps Across All Platforms and Devices



The GrapeCity family of products provides developers, designers, and architects the ultimate collection of easy-to-use tools for building sleek, high-performing, feature complete applications. With over 25 years of experience, we understand your needs and have developed a comprehensive line of products that includes innovative UI controls; cross-platform data visualization controls; and enterprise level reporting, analysis, and spreadsheet controls for Windows, web, and mobile platforms.

Learn more at:
componentsource.com/grapecity